

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIA EXATAS E TECNOLOGIA
GESTÃO DE QUALIDADE DE SOFTWARE

ILLGNER ANDERSON MORAES DE SOUZA
JOÃO VICTOR

SISTEMA FOREFY

ITACOATIARA-AM
2024

ILLGNER ANDERSON MORAES DE SOUZA
JOÃO VICTOR

SISTEMA FOREFY

Trabalho de Software apresentado como requisito parcial para obtenção de notas da disciplina Gestão da Qualidade de Software do curso de Engenharia de Software pela Universidade Federal do Amazonas.

ITACOATIARA - AM

2024

SUMÁRIO

INTRODUÇÃO	4
Requisitos Funcionais	5
Requisitos Não Funcionais	5
Prioridades dos Requisitos	5
Descrição Geral do Sistema	6
Descrição dos Usuários	6
DESENVOLVIMENTO	6
VERIFICAÇÃO - MÉTODO CHECKLIST	7
Checklist do modelo de negócios	7
Checklist da proposta de projeto de Software	8
Correção do Checklist Modelo de Negócio	8
Correção do Checklist Proposta de Projeto	9
Análise de Risco	10
Checklist do Modelo de Requisitos	11
Requisitos Funcionais	16
Requisitos Não-Funcionais	17
Checklist do Diagrama UML	18
DIAGRAMA DE CLASSE	19
DIAGRAMA DE SEQUÊNCIA	19
DIAGRAMA DE ATIVIDADE	21
Checklist do Diagrama UML	22
Checklist do Código-Fonte	22
Correção no Volume de Comentários	23
TESTE DE FUNCIONAL	26
TESTES UNITÁRIOS	31
TESTES DE SEGURANÇA	32
CONSIDERAÇÕES FINAIS	36

INTRODUÇÃO

O mundo no qual vivemos e suas diversas causas no aumento do efeito estufa, o desmatamento tem sido um dos maiores problemas ao redor do planeta, todo ano amazônia sofre com inúmeros desmatamentos ilegais em toda parte, afetando tanto a natureza como a vida das pessoas ao redor de nossa região.

Pensando neste específico problema, foi-se pensado no aplicativo Forefy. O aplicativo oferece uma solução para o problema de desmatamentos ilegais, onde através de um sensor de IoT conectado ao aplicativo, é possível notificar o usuário cadastrado se está havendo ou não um desmatamento na área específica onde o dispositivo está colocado.

A interpretação correta deste documento exige o conhecimento de termos específicos, que estão descritos abaixo.

Abreviações	Descrição
RF	Requisitos Funcionais
RFN	Requisitos Não Funcionais
RN	Regras de Negócios
UC	Use Cases (Casos de Uso)
LGDP	Lei Geral de Proteção de Dados

Requisitos Funcionais

Os requisitos funcionais descrevem tarefas e serviços que serão fornecidos pelo sistema aos seus usuários. Inclui também as interfaces externas do software.

Requisitos Não Funcionais

Requisitos não funcionais são restrições impostas sobre o software, que não se referem diretamente às funcionalidades do sistema. Eles podem ser relacionados aos requisitos funcionais, mas não dependem deles..

Prioridades dos Requisitos

- **Essencial:** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

- **Importante:** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável:** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

Descrição Geral do Sistema

O aplicativo Forefy é uma solução criativa para a detecção de desmatamento ilegal, funcionando como uma extensão de um dispositivo IoT capaz de detectar sons de motosserra, o aplicativo através do firebase recebe os dados dessa detecção e notifica o usuário sobre o possível desmatamento.

O Forefy utiliza o React Native junto do Firebase para criar uma plataforma simples e funcional de ser utilizada que junto ao dispositivo IoT consegue entregar uma solução para os problemas de desmatamento na região.

Descrição dos Usuários

O sistema contará com o usuário, que através de um sistema de cadastro-cliente poderá acompanhar de perto a localização do dispositivo IoT e ver se está ocorrendo desmatamento, além de poder adicionar contatos importantes e ligar para autoridades através de uma lista de contatos importantes.

DESENVOLVIMENTO

Aplicações (Teste de Verificação e Validação)

Natural no desenvolvimento de software, erros surgem em qualquer etapa do ciclo de vida do projeto. Diante disso, as etapas de verificação e validação de software se tornam essenciais e não podem ser vistas como uma atividade opcional, pois oferecem os métodos e ferramentas necessárias para garantir a qualidade de qualquer desenvolvimento.

O conceito da qualidade tem hoje importância fundamental para alavancar a competitividade das empresas. Atualmente, a preocupação com a qualidade deixou de ser um diferencial competitivo e passou a ser um pré-requisito básico para participação no mercado. No setor de software não é diferente. A disseminação do uso do software em todas as áreas, envolvendo monitoração, controle e gestão de

funções críticas, tem aumentado consideravelmente a importância da qualidade de software.

Qualidade hoje em dia não é apenas um diferencial de mercado para a empresa conseguir vender e lucrar mais, é um pré-requisito que a empresa deve conquistar para conseguir colocar seu produto no mercado global. Apesar da ideia de qualidade parecer aparentemente intuitiva, quando analisada com maior atenção, o conceito se revela um pouco mais complexo.

Na medida em que cresce a demanda por sistemas complexos, com grande responsabilidade no contexto das organizações, a qualidade desponta como um fator essencial no desenvolvimento de software, e cada vez mais há uma disposição de investimento nesta área. Entretanto, uma das primeiras dificuldades encontradas na definição e implantação de um programa de qualidade está em compreender o que, de fato, significa qualidade de software

VERIFICAÇÃO - MÉTODO CHECKLIST

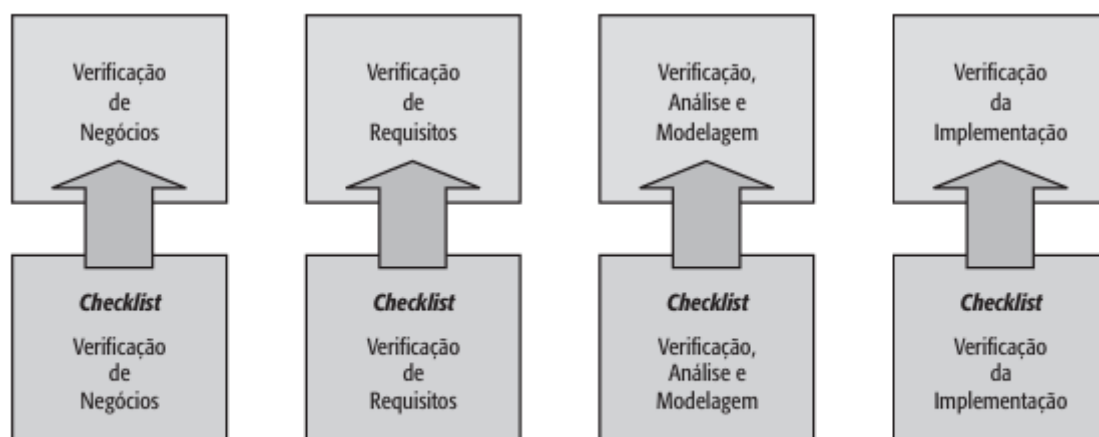


Figura 9.3 Checklist aplicada nas diversas fases dos testes de verificação

Checklist do modelo de negócios

Checklist do Modelo de Negócios		
Levantamento das Necessidades do Software		
- Todas as necessidades foram devidamente registradas.	Sim	Não
- Cada necessidade apontada possui uma descrição	Sim	Não
Definição das Características do Software		
- Cada característica atende ao menos a uma necessidade identificada.	Sim	Não

- Cada característica possui uma descrição clara.	Sim	Não
- Cada característica possui exemplos que auxiliam seu entendimento.	Sim	Não
- Existe uma rastreabilidade entre características e necessidades.	Sim	Não

Checklist da proposta de projeto de Software

Checklist da Proposta de Projeto de Software		
Definição dos Objetivos do Projeto		
- Todos os objetivos foram apontados e claramente descritos.	Sim	Não
- Todos os objetivos podem ser quantificáveis.	Sim	Não
- Todos os objetivos possuem data-limite para ocorrer.	Sim	Não
- Existe rastreabilidade entre objetivos e necessidades	Sim	Não
Definição dos Riscos		
- Todos os riscos foram identificados e adequadamente descritos.	Sim	Não
- Exige um plano de ação para risco definido.	Sim	Não
- Foram definidos “impacto” e “probabilidade” para cada risco apontado.	Sim	Não

Mostrar os defeitos encontrados e as correções

Correção do Checklist Modelo de Negócio

DESCRIÇÃO		
ID	NECESSIDADES	DESCRIÇÕES
NE-01	Adicionar Contatos	O sistema deve permitir que o usuário adicione contatos.
NE-02	Permitir Cadastro	O sistema deve permitir que os usuários se cadastrem e façam login no sistema.
NE-03	Permitir geolocalização do dispositivo de IoT	O sistema deve permitir a

		geolocalização do dispositivo de IoT.
NE-04	Permitir visualização do desmatamento em tempo real	O sistema deve permitir que os usuários visualizem se está ocorrendo desmatamento ou não
NE-05	Notificar ocorrência de desmatamento	O sistema deve notificar os usuários quando ocorrer um desmatamento.

RASTREABILIDADE CARACTERÍSTICAS E NECESSIDADES	
NECESSIDADES	CARACTERÍSTICAS
Adicionar Contatos	Funcionalidade, Usabilidade
Permitir Cadastro	Segurança, Confiabilidade
Permitir Geolocalização do Dispositivo de IoT	Portabilidade, Funcionalidade, Usabilidade
Permitir Visualização do Desmatamento em Tempo Real	Performance, Funcionalidade, Usabilidade
Notificar Ocorrência de Desmatamento	Operacionalidade

Correção do Checklist Proposta de Projeto

DESCRIÇÃO	
OBJETIVOS	DESCRIÇÕES
Levantamento das Necessidades e Requisitos	Recolher informações diretamente ligados a funcionalidade e operação do sistema, ligado diretamente a qualidade de software.
Codificação e Implementação	Desenvolver instruções usando linguagens de programação e dispositivos paralelos, instruções baseadas na documentação de requisitos.
Prototipagem e Testes	Representar graficamente o dispositivo e testar a consistência de suas funcionalidades
Implantação	Colocar o sistema em prática através de arquiteturas de software pré ou pós-estabelecidos.
Versionamento	Criar novas versões no código, assim que existir uma mudança significativa nele.

CRONOGRAMA QUANTIFICAÇÃO DOS OBJETIVOS						
OBJETIVOS	1° MÊS	2° MÊS	3° MÊS	4° MÊS	5° MÊS	6° MÊS
Levantamento das						

Necessidades e Requisitos						
Codificação e Implementação						
Prototipagem e Testes						
Implantação						
Versionamento						

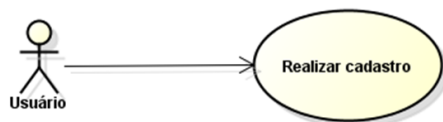
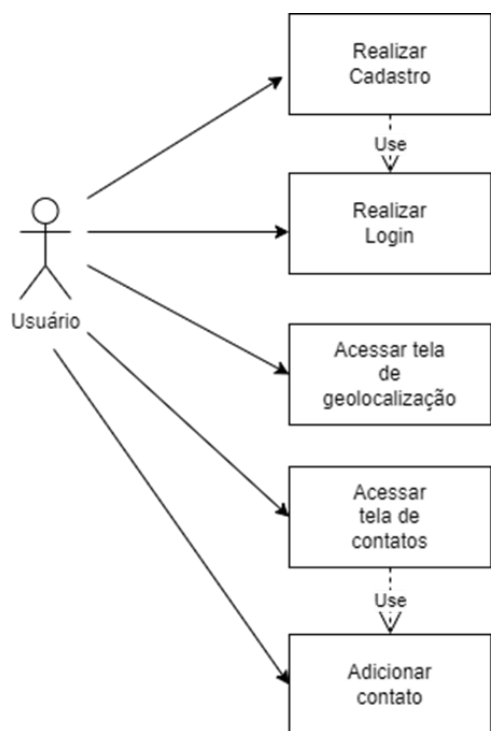
RASTREABILIDADE OBJETIVOS E NECESSIDADES	
OBJETIVOS	NECESSIDADES
Levantamento das Necessidades e Requisitos	NE-01, NE-02, NE-03, NE-04, NE-05.
Codificação e Implementação	NE-01, NE-02, NE-03, NE-04, NE-05.
Prototipagem e Testes	NE-01, NE-02, NE-03, NE-04, NE-05.
Implantação	NE-01, NE-02, NE-03, NE-04, NE-05.
Versionamento	NE-01, NE-02, NE-03, NE-04, NE-05.

Análise de Risco

Descrição	Exemplo	Impacto	Resposta ao Risco
Risco de segurança	O aplicativo ser comprometido e ser exposto informações confidenciais dos usuários.	Qualidade, segurança e confiabilidade.	Durante e após o desenvolvimento deverá ser testado se há vulnerabilidades de segurança e aplicar patches e <u>updates</u> para corrigi-las.

Checklist do Modelo de Requisitos

Checklist do Modelo de Requisitos		
Diagrama de Casos de Uso		
- Existe um modelo de casos de uso para cada subsistema identificado.	Sim	Não
- Todos os casos de uso estão adequadamente descritos.	Sim	Não
- Todos os atores estão adequadamente representados.	Sim	Não
Levantamento de Requisitos		
- Cada caso de uso representa um requisito funcional.	Sim	Não
- Exige rastreabilidade entre requisitos identificados e necessidades.	Sim	Não
- Requisitos foram avaliados por importância, volatilidade e criticidade.	Sim	Não
Especificações Funcionais		
- Cada requisito funcional possui uma especificação detalhada.	Sim	Não
- As especificações contemplam os fluxos básicos, alternativos e exceção.	Sim	Não
- As especificações contemplam pré-requisitos e pós-condições.	Sim	Não
Especificações Não Funcionais		
- Todas as categorias de requisitos não funcionais foram levantadas.	Sim	Não
- Cada requisito não funcional possui uma especificação detalhada.	Sim	Não
- Todas as dependências dos componentes foram estabelecidas.	Sim	Não



Subsistema	Use Case 01 - Realizar cadastro				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
UC 01	Realizar cadastro	Realizar cadastro para que se possa obter acesso com as totalidades e funcionalidades do sistema.	As ações fazem referências às tabelas.	RF01	<u>usuário</u>

Fluxos de Eventos Normais

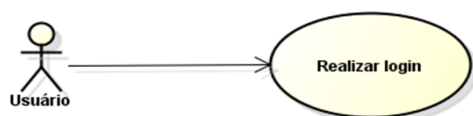
Nome do Fluxo de Eventos Normal	Pré-condição	Descrição
Realizar cadastro	O usuário precisa entrar no aplicativo e clicar no botão de cadastrar e cadastrar no sistema para poder usá-lo.	<ol style="list-style-type: none"> 1. O usuário entra no aplicativo. 2. O usuário entra na tela de cadastro. 3. O sistema exibe tela de cadastro. 4. O usuário insere os dados necessários para realizar o cadastro. 5. O cadastro do cliente é confirmado e ele está pronto para acessar o aplicativo.

Fluxos de Eventos Variantes

Nome do Fluxo de Eventos Relacionado	Variante Normal	Descrição
Realizar cadastro	O usuário seleciona opção de se cadastrar, porém já possui cadastro.	<ol style="list-style-type: none"> 1. O usuário acessa a tela de cadastrar. 2. O sistema exibe a tela de cadastro. 3. O usuário preenche a tela de cadastro. 4. O sistema informa que já existe uma conta conectada ao <u>email</u> ou <u>email</u> é inválido.

Fluxos de Eventos de Exceção

Nome do Fluxo de Eventos Relacionado	Condição de Exceção	Descrição
Realizar cadastro	O sistema valida informações - Campos obrigatórios não preenchidos	<ol style="list-style-type: none"> 1. O usuário acessa a tela de cadastrar. 2. O sistema exibe a tela de cadastro. 3. O usuário preenche a tela de cadastro. 4. O sistema não consegue validar as informações com sucesso.



Subsistema	Use Case 02 - Realizar <u>login</u>				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
UC 02	Realizar <u>login</u>	Realizar <u>login</u> para que se possa obter acesso com as totalidades e funcionalidades do sistema.	As ações fazem referências às tabelas.	RF01	<u>usuário</u>

Fluxos de Eventos Normais

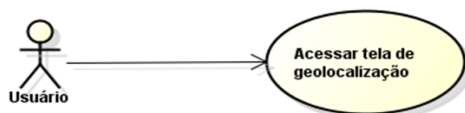
Nome do Fluxo de Eventos Normal	Pré-condição	Descrição
Realizar <u>login</u> de usuário	O usuário precisa entrar no aplicativo e clicar na tela de <u>login</u> com seu nome de usuário e senha.	<ol style="list-style-type: none"> 1. O usuário entra no aplicativo. 2. O usuário entra na tela de <u>login</u>. 3. O sistema exibe tela de <u>login</u>. 4. O usuário insere os dados necessários para realizar o <u>login</u>. 5. O sistema informa a conclusão do <u>login</u>. 6. O usuário é redirecionado a sua respectiva página.

Fluxos de Eventos Variantes

Nome do Fluxo de Eventos Normal Relacionado	Variante	Descrição
Realizar <u>login</u> de usuário	O usuário tenta entrar no aplicativo, porém não possui cadastro.	<ol style="list-style-type: none"> 1. O usuário acessa a tela de <u>login</u>. 2. O sistema exibe a tela de <u>login</u>. 3. O usuário preenche a tela de <u>login</u>. 4. O sistema não valida o <u>login</u>. 5. O sistema apresenta a mensagem "Dados incorretos ou usuário inexistente".

Fluxos de Eventos de Exceção

Nome do Fluxo de Eventos Normal Relacionado	Condição de Exceção	Descrição
Realizar <u>login</u> de usuário	O sistema não valida o <u>login</u> , pois o usuário ou senha estão incorretos.	<ol style="list-style-type: none"> 1. O usuário acessa a tela de <u>login</u>. 2. O sistema exibe a tela de <u>login</u>. 3. O usuário preenche a tela de <u>login</u>. 4. O sistema não valida o <u>login</u>. 5. O sistema apresenta a mensagem "Dados incorretos ou usuário inexistente".



Subsistema	Use Case 03 - Acessar tela de relatórios				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
UC 03	Acessar tela de <u>geolocalização</u>	Acessar tela de <u>geolocalização</u> para que se possa ver a localização do dispositivo.	As ações fazem referências às tabelas.	RF02, RF03, RF05	<u>usuário</u>

Fluxos de Eventos Normais

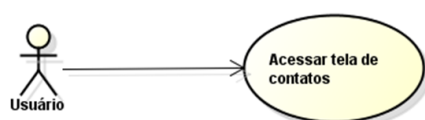
Nome do Fluxo de Eventos Normal	Pré-condição	Descrição
Acessar tela de <u>geolocalização</u>	O usuário acessa a tela de <u>geolocalização</u> e ver a localização do dispositivo <u>IoT</u> .	<ol style="list-style-type: none"> 1. O usuário entra na tela de <u>geolocalização</u>. 2. O usuário pode agora consultar seus relatórios e ver a validade dos mesmos.

Fluxos de Eventos Variantes

Nome do Fluxo de Eventos Normal Relacionado	Variante	Descrição
Acessar tela de <u>geolocalização</u>	Nenhum.	Nenhum.

Fluxos de Eventos de Exceção

Nome do Fluxo de Eventos Relacionado	Condição de Exceção	Descrição
Acessar tela de <u>geolocalização</u>	O usuário é levado para a tela de <u>login</u> porém não consegue abrir a tela de <u>geolocalização</u> por problema de conexão de rede.	<ol style="list-style-type: none"> O usuário entra na tela de <u>geolocalização</u>. O usuário não consegue abrir a tela por problema de conexão.



Subsistema	Use Case 04 – Adicionar Contatos				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
UC 04	Acessar contatos	Acessar a tela de contatos	As ações fazem referências às tabelas.	RF04	<u>usuário</u>

Fluxos de Eventos Normais

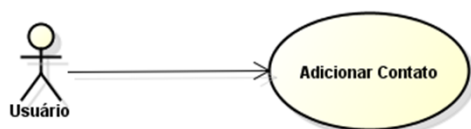
Nome do Fluxo de Eventos Normal	Pré-condição	Descrição
Acessar a tela de contatos	O usuário entra na tela de contatos	<ol style="list-style-type: none"> O usuário clica no ícone da tela de contatos. Ele(a) é redirecionado a tela de contatos.

Fluxos de Eventos Variantes

Nome do Fluxo de Eventos Normal	Variante	Descrição
Acessar a tela de contatos	Nenhum.	Nenhum.

Fluxos de Eventos de Exceção

Nome do Fluxo de Eventos Normal	Condição de Exceção	Descrição
Acessar a tela de contatos	Nenhum.	Nenhum.



Subsistema	Use Case 05 – Adicionar Contatos				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
UC 05	Adicionar contatos	Adicionar contatos	As ações fazem referências às tabelas.	RF04	<u>usuário</u>

Fluxos de Eventos Normais

Nome do Fluxo de Eventos Normal	Pré-condição	Descrição
Adicionar contatos	O usuário pode adicionar contatos importantes no app	<ol style="list-style-type: none"> 1. Na tela de relatórios o usuário clica em um botão de adicionar contato. 2. O usuário cadastra um contato importante.

Fluxos de Eventos Variantes

Nome do Fluxo de Eventos Normal Relacionado	Variante	Descrição
Adicionar contatos	Nenhum.	Nenhum.

Fluxos de Eventos de Exceção

Nome do Fluxo de Eventos Normal Relacionado	Condição de Exceção	Descrição
Adicionar contatos	Nenhum.	Nenhum.

Requisitos Funcionais

Identificador	Descrição	Prioridade	Requisitos Relacionados
RF01	O sistema deve permitir que os usuários se cadastrem e façam login no sistema.	Essencial	
RF02	O sistema deve permitir que os usuários visualizem se está ocorrendo desmatamento ou não	Essencial	RF01, RF03
RF03	O sistema deve notificar os usuários quando ocorrer um desmatamento	Importante	RF01, RF02
RF04	O sistema deve permitir que o usuário adicione contatos importantes	Importante	RF01
RF05	O sistema deve permitir a geolocalização do dispositivo de IoT	Essencial	RF01, RF02, RF03

Requisitos Não-Funcionais

Identificador	Descrição	Categoria	Escopo	Prioridade	Requisitos Relacionados
RNF01	O sistema deve ser seguro contra acesso não autorizado.	Segurança	Geral	Essencial	RN01, RN02, RN03, RN04, RN05, RNF02
RNF02	O sistema deve estar disponível 24 horas por dia, 7 dias por semana.	Disponibilidade	Geral	Essencial	RN01, RN02, RN03, RN04, RN05, RNF02
RNF03	O sistema deve ser fácil de usar e aprender.	Usabilidade	Geral	Essencial	RN01, RN02, RN03, RN04, RN05, RNF02
RNF04	O sistema deve ser compatível com as versões mais recentes do <u>Android</u> e <u>IOS</u> .	Compatibilidade	Geral	Essencial	RN01, RN02, RN03, RN04, RN05,
RNF05	O sistema deve ser compatível com a versão mais recente do <u>Firebase</u> .	Compatibilidade	<u>Backend</u>	Essencial	RN01, RN02, RN03, RN04, RN05, RNF02

Checklist do Diagrama UML

Checklist do Modelo de Requisitos		
Diagrama de Classes		
- Todas as classes possuem nome e descrição adequados.	Sim	Não
- Todos os atributos das classes possuem nome e descrição adequados.	Sim	Não
- Todos os serviços da classe possuem nome e descrição adequados.	Sim	Não
Diagrama de Estado		
- Todas as transições de estado possuem um serviço ou evento associado.	Sim	Não
- Todos os estados possuem nome e descrição adequados.	Sim	Não
- Todas as transições de estado refletem o real ciclo de vida da classe.	Sim	Não
Diagrama de Componentes		
- Os <i>packpages</i> agrupam componentes com as mesmas características.	Sim	Não
- Cada componente agrupa classes de única camada: <i>user</i> , <i>business</i> , <i>data</i> .	Sim	Não
- Todas as dependências dos componentes foram estabelecidas.	Sim	Não

NÃO TEMOS DIAGRAMAS DE ESTADO E DIAGRAMAS DE COMPONENTES. APESAR DO DIAGRAMA DE CLASSES ESTÁ DE FORMA REDUNDANTE, TEMOS A DISPOSIÇÃO OS DIAGRAMAS DE SEQUÊNCIA E ATIVIDADES PARA VISUALIZAÇÃO DOS FLUXOS DO SISTEMA FOREFY.

DIAGRAMA DE CLASSE

Os diagramas de classe descrevem o que deve estar presente no sistema a ser modelado, é um modelo padronizado para descrever uma abordagem de programação orientada ao objeto.

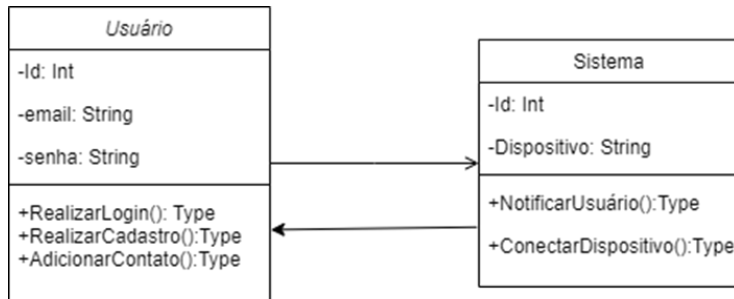


DIAGRAMA DE SEQUÊNCIA

Os diagramas de sequência são descritos como uma espécie de diagrama de interação, em que descreve como um grupo de objetos trabalha, além de expor em qual ordem. Este diagrama tem por objetivo entender a necessidade de um novo sistema ou para documentar um processo existente

DIAGRAMA DE SEQUÊNCIA: REALIZAR CADASTRO

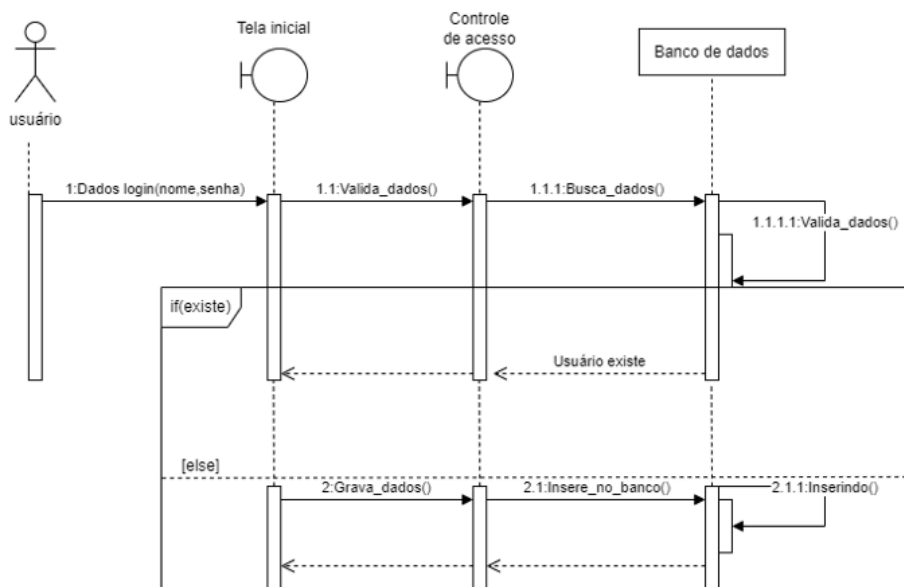


DIAGRAMA DE SEQUÊNCIA: REALIZAR LOGIN

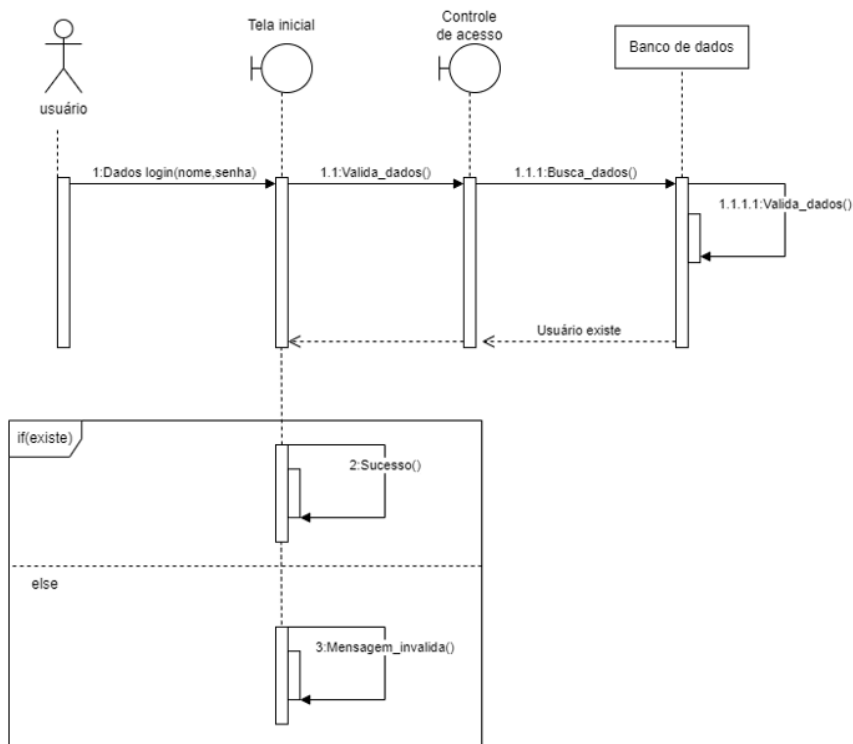


DIAGRAMA DE SEQUÊNCIA: ACESSAR TELA DE GEOLOCALIZAÇÃO

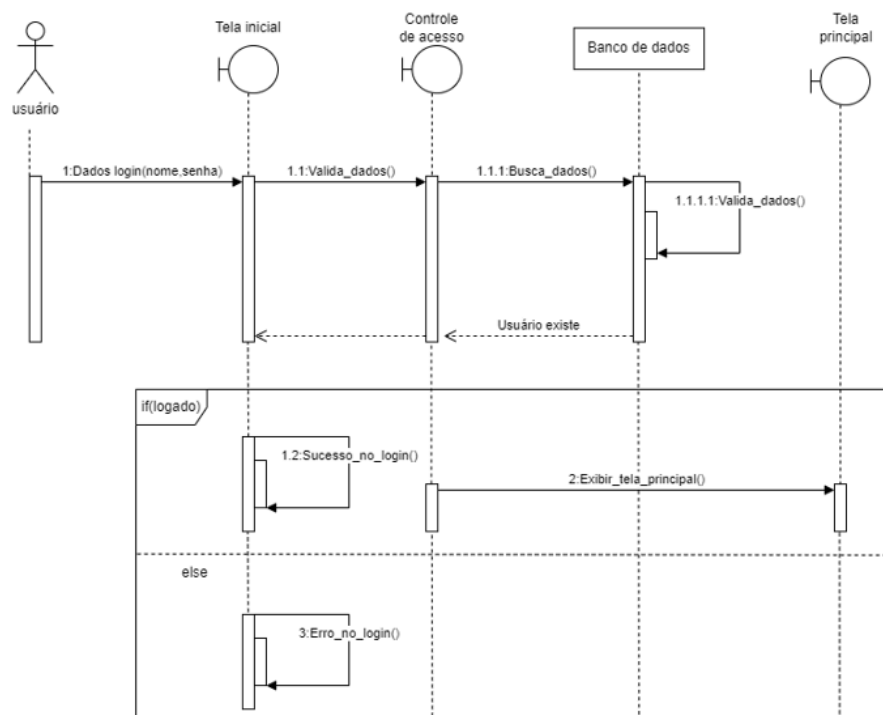


DIAGRAMA DE ATIVIDADE

Um diagrama de atividade é uma representação gráfica que descreve o fluxo de controle ou o fluxo de atividades em um sistema, processo ou procedimento. Esses diagramas são usados para modelar o comportamento dinâmico de um sistema, enfatizando as atividades executadas em um determinado período de tempo e como essas atividades estão interconectadas.

DIAGRAMA DE ATIVIDADE: REALIZAR CADASTRO

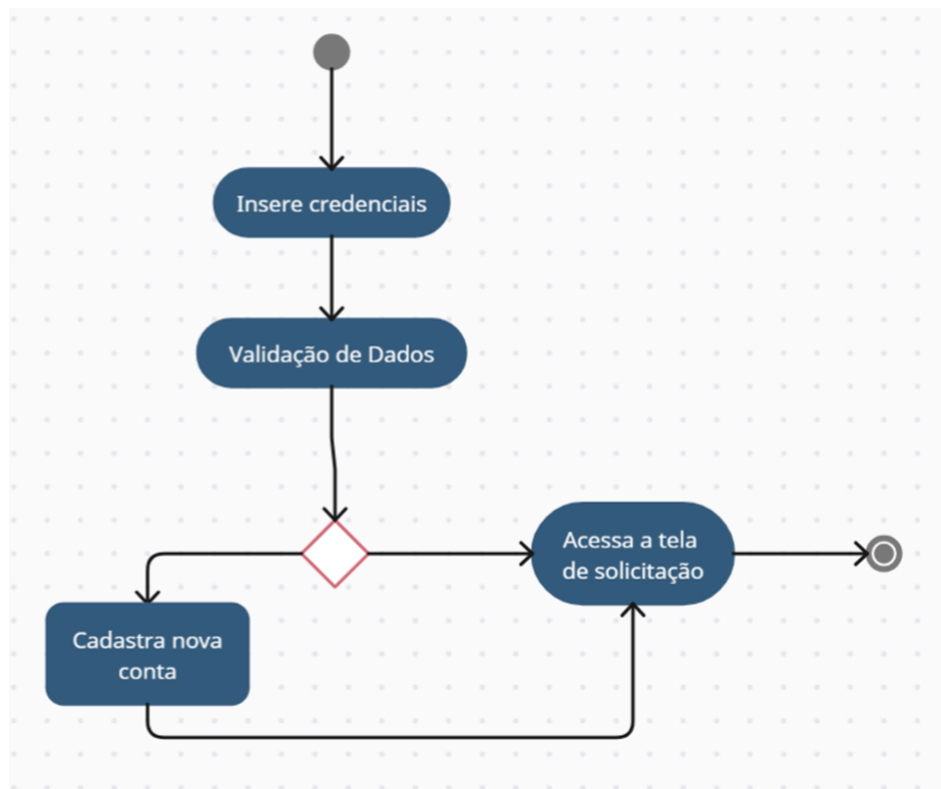
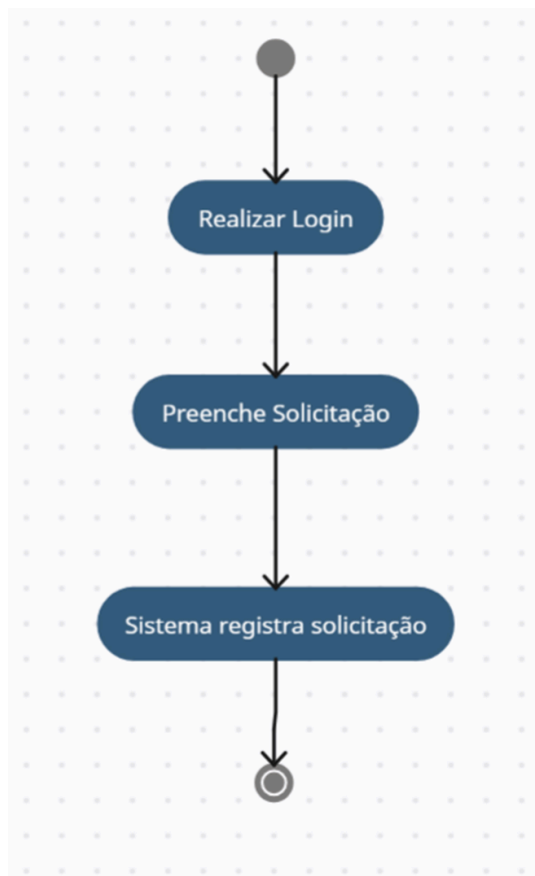


DIAGRAMA DE ATIVIDADE: REALIZAR LOGIN



Checklist do Diagrama UML

Checklist da Arquitetura		
Suportar Mudanças nos Negócios		
- Existem parametrizações que modificam a funcionalidade da aplicação.	Sim	Não
Suportar Mudanças Tecnológicas		
- O software possui independência do banco de dados.	Sim	Não
- O software possui independência do sistema operacional.	Sim	Não
- Todas as transições de estado refletem o real ciclo de vida da classe.	Sim	Não

Checklist do Código-Fonte

Checklist do Código Fonte
Comparação do Código-Fonte

- Todas as classes do modelo foram implementadas.	Sim	Não
- Todos os métodos de cada classe foram implementados.	Sim	Não
- Todos os atributos de cada classe foram implementados.	Sim	Não
Mensagens Apresentadas ao Usuário Final		
- Nenhuma mensagem apresenta erros gramaticais.	Sim	Não
- Todas as mensagens são claras e bem objetivas.	Sim	Não
- Todas as mensagens apresentam ícones adequados ao contexto.	Sim	Não
Legibilidade do Código		
- Todas as estruturas estão adequadamente indentadas.	Sim	Não
- Não existem linhas agrupadas com IF, SELECT, FOR NEXT E FOR EACH.	Sim	Não
- Todas as declarações de variáveis e constantes estão no início da rotina.	Sim	Não
- Não existem vários comandos em uma única linha.	Sim	Não
Volume de Comentários		
- Todas as rotinas possuem descrição sobre seu comportamento.	Sim	Não
- Todos os desvios de rotinas possuem um comentário	Sim	Não

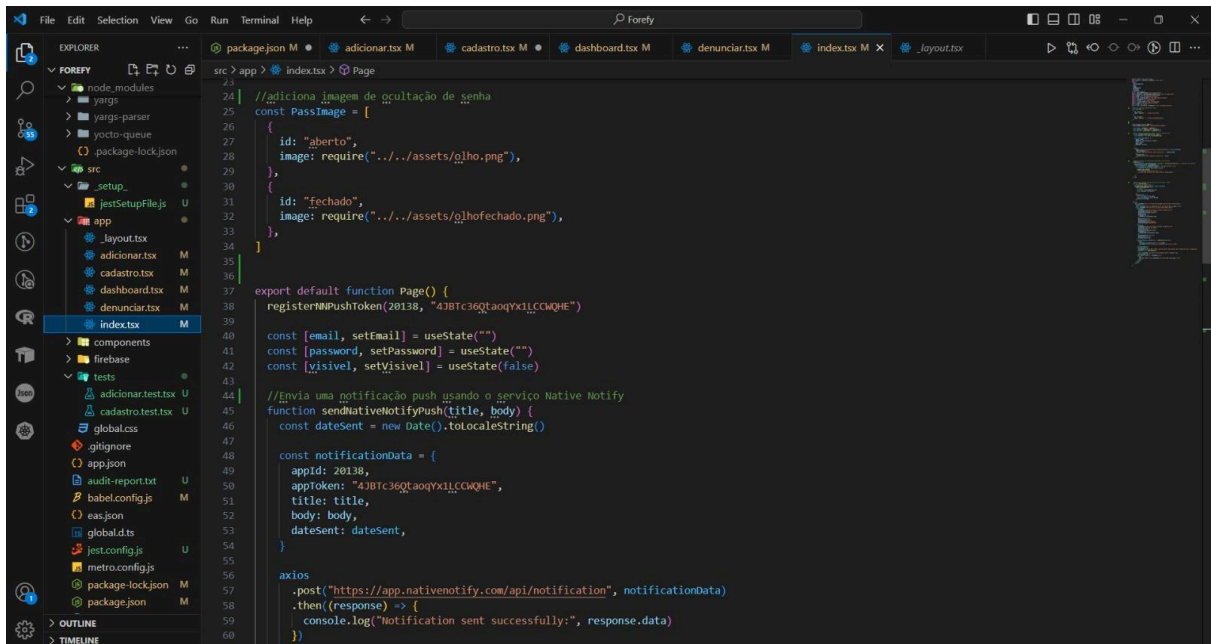
Correção no Volume de Comentários

```
13
14 //Adiciona nome, setor, cargo e numero para lista de contatos
15 export default function Adicionar() {
16   const [name, setName] = useState("")
17   const [setor, setSetor] = useState("")
18   const [cargo, setCargo] = useState("")
19   const [numero, setNumero] = useState("")
20
21   const addContact = async (event) => {
22     event.preventDefault()
23     const numeroFormatted = numero.replace(/\\/g, "")
24     if (numeroFormatted.length !== 11 || !/^[0-9]+$/.test(numeroFormatted)) {
25       alert("O numero deve conter exatamente 11 digitos.")
26       return
27     }
28     if (!/^[A-Za-z\\s]{6,}$/.test(name)) {
29       alert("O nome deve conter apenas letras e ter pelo menos 6 caracteres.")
30       return
31     }
32     try {
33       const logCollectionRef = collection(db, "contatos")
34       const user = auth.currentUser
35       const logDocRef = doc(logCollectionRef)
36       await setDoc(logDocRef, {
37         contatoId: logDocRef.id,
38         addedBy: user.uid,
39         contatoNome: name,
40         contatoSetor: setor,
41         contatoCargo: cargo,
42         contatoNumero: numero,
43       })
44       alert("contato adicionado com sucesso!")
45       setName("")
46       setCargo("")
47       setSetor("")
48       setNumero("")
49     } catch (error) {
50       alert(
```

```
src > app > cadastro.tsx > Cadastro
You, 7 minutes ago | 2 authors (LucasBOC and one other)
1 import { View, Text, TouchableOpacity, TextInput, Image } from "react-native"
2 import { Link } from "expo-router"
3 import { useState } from "react"
4 import { createUserWithEmailAndPassword } from "firebase/auth"
5 import { auth } from "@firebase/firebase"
6 import { router } from "expo-router"
7
8 import Header from "@components/model/header"
9 import Input from "@components/ui/input"
10
11 //adiciona imagem de ocultação de senha
12 const PassImage = [
13   {
14     id: "aberto",
15     image: require("../assets/olho.png"),
16   },
17   {
18     id: "fechado",
19     image: require("../assets/olhofechado.png"),
20   },
21 ]
22
23 //cadstra usuário à plataforma
24 export default function cadastro() {
25   const [email, setEmail] = useState("")
26   const [password, setPassword] = useState("")
27   const [visivel, setVisivel] = useState(false)
28
29   function firebaseSignUp() {
30     createUserWithEmailAndPassword(auth, email, password)
31       .then((userCredential) => {
32         // Signed up
33         const user = userCredential.user
34         console.log(user)
35         router.push("/")
36         // ...
37       })
38   }
39 }
```

```
7 import Header from "@components/model/header"
8 import Footer from "@components/model/footer"
9
10 //script para posição fixa de onde poderia estar o dispositivo
11 export default function Dashboard() {
12   const [mapRegion, setMapRegion] = useState({
13     latitude: -3.1175289773523414,
14     longitude: -58.44310195614139,
15     latitudeDelta: 0.0922,
16     longitudeDelta: 0.0421
17   })
18
19   return (
20     <View className="flex flex-1 bg-zinc-100 items-center justify-center">
21       <Header />
22       <MapView className="h-16 w-16" style={{ width: 350, height: 350 }} mapType="mutedStandard" region={mapRegion}>
23         <Marker coordinate={mapRegion} title="Marker" />
24       </MapView>
25       <View className="w-16 h-16 bg-blue-500" />
26       <Text className="text-2xl font-medium">Latitude: {props.latitude}</Text>
27       <Text className="text-2xl font-medium">Longitude: {props.longitude}</Text>
28     </View>
29   )
30 }
31
32 //Interface das coordenadas
33 interface CoordenadasProps {
34   latitude: number
35   longitude: number
36 }
37
38 function Coordenadas(props: CoordenadasProps) {
39   return (
40     <View className="mt-10">
41       <Text className="text-2xl font-medium">Localização:</Text>
42       <Text className="text-2xl font-medium">Latitude: {props.latitude}</Text>
43       <Text className="text-2xl font-medium">Longitude: {props.longitude}</Text>
44     </View>
45   )
46 }
```

```
1 import React, { useEffect, useRef, useState } from "react"
2 import { View, Text, ScrollView, SafeAreaView } from "react-native"
3 import { Link } from "expo-router"
4
5 import Header from "@components/model/header"
6 import Footer from "@components/model/footer"
7 import Contato from "@components/ui/contato"
8
9 import { collection, getDocs, query } from "firebase/firestore"
10 import { db } from "@firebase/firebase"
11
12 //função para denunciar caso de queimada
13 export default function Denunciar() {
14   const [contatos, setContatos] = useState([])
15   const [isLoading, setIsLoading] = useState(true)
16
17   const fetchContatos = async () => {
18     try {
19       const contatosCollection = collection(db, "contatos")
20       const q = query(contatosCollection)
21
22       const querySnapshot = await getDocs(q)
23       const fetchedContatos = querySnapshot.docs.map((doc) => ({
24         id: doc.id,
25         ...doc.data(),
26       }))
27       setContatos(fetchedContatos)
28     } catch (error) {
29       console.error("Error fetching contatos:", error)
30     }
31     finally {
32       setIsLoading(false)
33     }
34   }
35
36   useEffect(() => {
37     fetchContatos()
38   }, [])
39 }
```

VALIDAÇÃO

TESTE DE FUNCIONAL

22:23

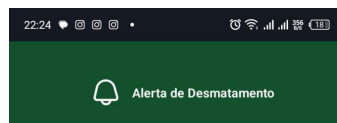
Alerta de Desmatamento

FAÇA O CADASTRO:

Criar Conta

Já tem conta? [Entre aqui!](#)

Criar Conta - Ao criar a conta o app não loga diretamente e sim leva para a tela de login



FAÇA O LOGIN:

teste@email.com

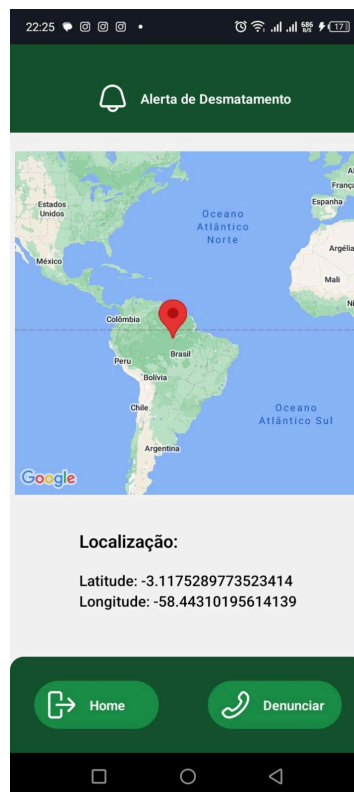
***** 

Entrar

Não tem conta? [Crie aqui!](#)

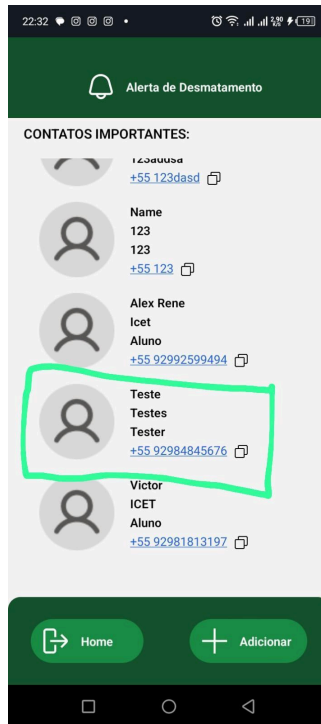


Tela de login - ao colocar a senha e email certos levam diretamente a tela do dashboard



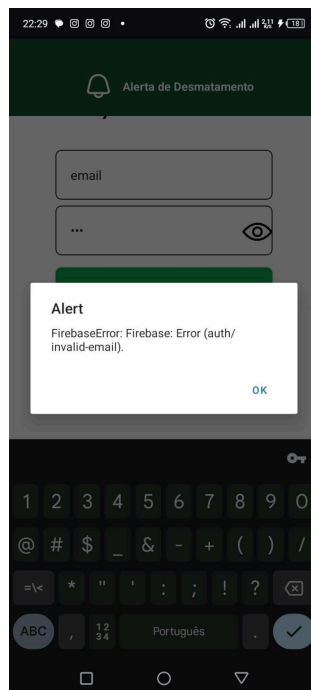
Dashboard - Tela de geolocalização

Ao criar um contato novo é dada a mensagem de contato adicionado com sucesso, porém não leva diretamente a tela de contatos.



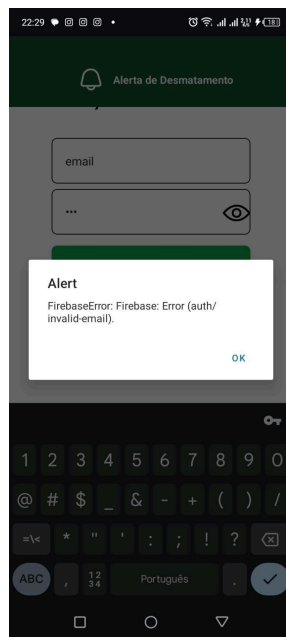
Tela Contato:

Ao voltar a tela de contatos o contato novo é atualizado com sucesso

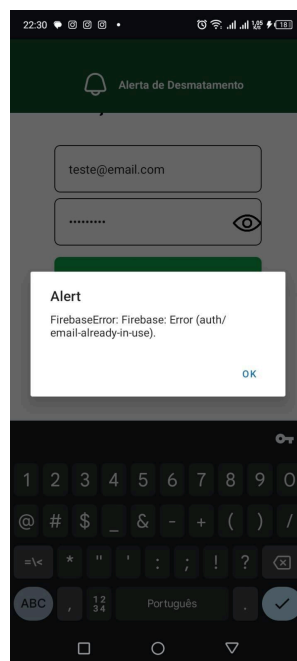


Erro de cadastro de usuário:

Ao tentar colocar um email inválido é mostrado uma tela de erro com email inválido.



Erro de cadastro de usuário:
Ao tentar colocar um email inválido é mostrado uma tela de erro com email inválido.



Teste com email em uso:
Ao tentar criar uma conta com um email já cadastrado mostra que o email já está em uso.

TESTES UNITÁRIOS

FERRAMENTA DE TESTE APLICADO = JEST FRAMEWORK

O Jest pode ser instalado pelo Yarn ou pelo NPM como dependência externa. Comando para adicionar o framework Jest:

yarn = yarn add - -dev jest

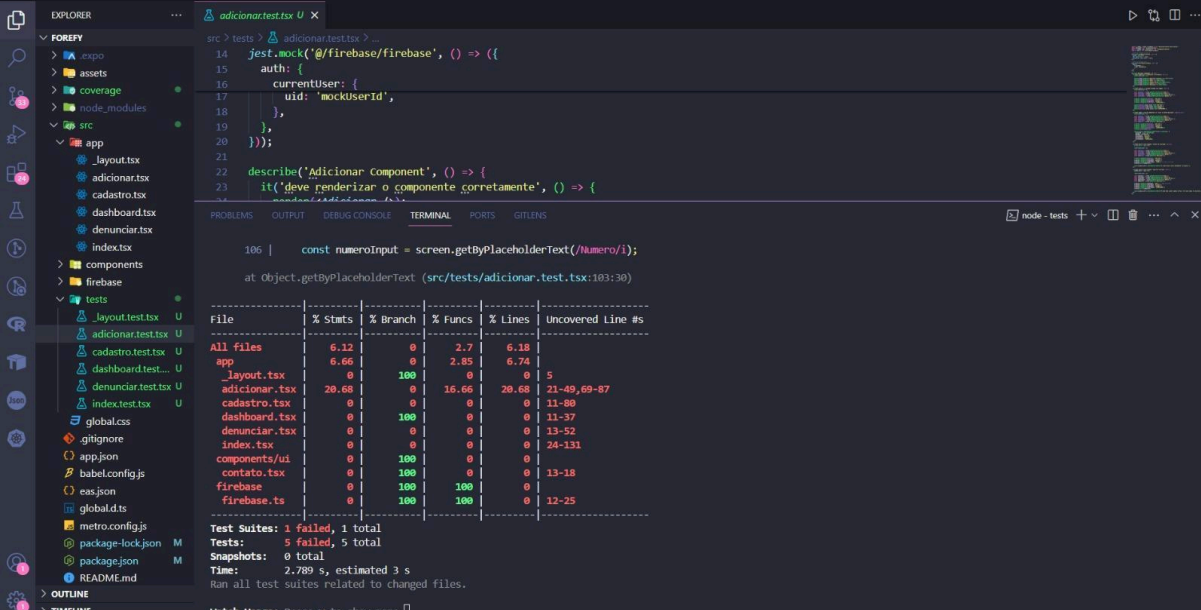
npm = npm install - -save-dev jest

Configuração

O Jest utiliza o próprio gerenciador de pacotes usado em sua instalação para executar os testes através do comando test, para isso é necessário adicionar ao documento package.json a seguinte sessão:

```
{  
  "scripts": {  
    "test": "jest"  
  }  
}
```

Falhas nos testes unitários aplicados em Jest



The screenshot shows a VS Code interface with a project structure on the left, a test file in the center, and a terminal window at the bottom displaying Jest test results. The test results table is as follows:

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	6.12	0	2.7	6.18	
app	6.66	0	2.85	6.74	
_layout.tsx	0	100	0	0	5
adicionar.tsx	20.68	0	16.66	20.68	21-49,69-87
cadastro.tsx	0	0	0	0	11-80
dashboard.tsx	0	100	0	0	11-37
denunciar.tsx	0	0	0	0	13-52
index.tsx	0	0	0	0	24-131
components/ui	0	100	0	0	
contato.tsx	0	100	0	0	13-18
firebase	0	100	100	0	
firebase.ts	0	100	100	0	12-25

Test Suites: 1 failed, 1 total
Tests: 5 failed, 5 total
Snapshots: 0 total
Time: 2.789 s, estimated 3 s
Run all test suites related to changed files.

A função `.toBe(valor)` testa se o valor passado é idêntico ao esperado em valor e tipo.

A função `.toEqual(valor)` testa recursivamente cada valor do objeto ou array.

A função `toBeNull` Testa se o resultado passado tem valor igual a null.

A função `toBeUndefined` testa se o resultado passado tem valor igual a undefined.

A função `toBeDefined` testa se o resultado passado não tem valor igual a `undefined`.

A função `toBeTruthy` testa se o resultado passado tem valor que pode ser passado como `true` em um `if`.

A função `toBeFalsy` testa se o resultado passado tem valor que pode ser passado como `false` em um `if`.

A função `toBeGreaterThan` testa se o resultado passado é maior que o esperado.

A função `toBeGreaterThanOrEqual` testa se o resultado passado é maior ou igual ao esperado.

A função `toBeLessThan` testa se o resultado passado é menor que o esperado.

A função `toBeLessThanOrEqual` testa se o resultado passado é menor ou igual ao esperado.

TESTES DE SEGURANÇA

Testes de segurança são essenciais para garantir a proteção de dados e informações em sistemas, aplicativos e redes. Eles envolvem a avaliação de possíveis vulnerabilidades e pontos de falha que podem ser explorados por invasores e hackers mal-intencionados, comprometendo a integridade, disponibilidade e confidencialidade dos dados.

`grpc/grpc-js`

Vulnerabilidade: O `@grpc/grpc-js` pode alocar memória para mensagens recebidas acima dos limites configurados, o que pode levar a ataques de DoS (negação de serviço).

`braces <3.0.3`

Vulnerabilidade: O `braces` tem um problema de consumo incontrolado de recursos. Um atacante pode explorar isso para consumir recursos do sistema e causar uma negação de serviço.

`fast-loops <1.1.4`

Vulnerabilidade: O `fast-loops` é vulnerável à poluição de protótipos, que pode permitir que um atacante modifique o protótipo de um objeto global e afete o comportamento do código.

`fast-xml-parser <4.4.1`

Vulnerabilidade: O `fast-xml-parser` é vulnerável a ataques de ReDOS (Recursos Denial of Service) na análise de moedas. Isso pode levar a altos tempos de processamento e negação de serviço.

follow-redirects <=1.15.5

Vulnerabilidade: O follow-redirects pode manter o cabeçalho Proxy-Authorization em redirecionamentos entre hosts, o que pode levar a vazamento de informações.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\victo\Music\Forefy> npm audit
# npm audit report

@grpc/grpc-js 1.9.0 - 1.9.14
Severity: moderate
@grpc/grpc-js can allocate memory for incoming messages well above configured limits - https://github.com/advisories/GHSA-7v5v-9h63-cj86
fix available via `npm audit fix`
node_modules/@grpc/grpc-js

braces <3.0.3
Severity: high
Uncontrolled resource consumption in braces - https://github.com/advisories/GHSA-grv7-fg5c-xmjg
fix available via `npm audit fix`
node_modules/braces

fast-loops <1.1.4
Severity: high
robinwaser fast-loops vulnerable to prototype pollution - https://github.com/advisories/GHSA-3q56-9cc2-46j4
fix available via `npm audit fix`
node_modules/fast-loops

fast-xml-parser <4.4.1
Severity: high
fast-xml-parser vulnerable to ReDOS at currency parsing - https://github.com/advisories/GHSA-mpg4-rc92-vx8v
fix available via `npm audit fix`
node_modules/fast-xml-parser

follow-redirects <=1.15.5
Severity: moderate
follow-redirects' Proxy-Authorization header kept across hosts - https://github.com/advisories/GHSA-cxjh-pqwp-8mfp
fix available via `npm audit fix`
node_modules/follow-redirects
```

ip

Vulnerabilidade: A biblioteca ip tem uma categorização incorreta em isPublic, o que pode levar a SSRF (Server-Side Request Forgery).

```
ip *
Severity: high
ip SSRF improper categorization in isPublic - https://github.com/advisories/GHSA-2p57-rm9w-gvfp
fix available via `npm audit fix --force`
Will install react-native@0.73.9, which is outside the stated dependency range
node_modules/ip
  @react-native-community/cli-doctor <=10.2.5 || 11.0.0-alpha.0 - 11.4.0 || 12.0.0-alpha.0 - 12.3.5 || 13.0.0 - 13.5.2
    Depends on vulnerable versions of ip
    node_modules/@react-native-community/cli-doctor
      @react-native-community/cli 4.13.0 - 9.3.5 || 11.0.2 - 11.4.0 || 12.0.0-alpha.0 - 12.3.5 || 13.0.0 - 13.5.2
        Depends on vulnerable versions of @react-native-community/cli-doctor
        Depends on vulnerable versions of @react-native-community/cli-hermes
        node_modules/@react-native-community/cli
          react-native <=0.0.0-20221027-2059-ba5454c42 || 0.69.0-rc.0 - 0.70.15 || 0.72.0-rc.0 - 0.72.10 || 0.73.0-nightly-20230506-1af868c52 - 0.73.4 || 0.74.0-nightly-20231003-5dedf277c - 0.74.0-rc.9 ||
          >=0.75.0-nightly-20240221-a1171f79f
            Depends on vulnerable versions of @react-native-community/cli
            node_modules/react-native
              @react-native-community/cli-hermes <=10.2.0 || 11.0.0-alpha.0 - 11.4.0 || 12.0.0-alpha.0 - 12.3.5 || 13.0.0 - 13.5.2
                Depends on vulnerable versions of ip
                node_modules/@react-native-community/cli-hermes
```

semver 7.0.0 - 7.5.1

Vulnerabilidade: O semver é vulnerável a ataques de ReDOS devido a uma expressão regular ineficiente. Isso pode levar a altos tempos de processamento e negação de serviço.

Fix: Atualize para uma versão segura usando npm audit fix --force. Isso pode instalar uma versão diferente de expo-notifications, o que é uma mudança significativa.


```

semver 7.0.0 - 7.5.1
Severity: moderate
semver vulnerable to Regular Expression Denial of Service - https://github.com/advisories/GHSA-C2qf-rxjj-qgqW
fix available via 'npm audit fix --force'
Will install expo-notifications@0.28.12, which is a breaking change
node_modules/@expo/image-utils/node_modules/semver
  @expo/image-utils <=0.0.1-canary-20240109-93608d8 || 0.3.10-alpha.0 - 0.4.2
  Depends on vulnerable versions of semver
node_modules/@expo/image-utils
  @expo/cli <=0.0.0-canary-20231123-1b19f96-4 || 0.0.1-canary-20231125-d60ee44 - 0.18.0
  Depends on vulnerable versions of @expo/image-utils
  Depends on vulnerable versions of @expo/prebuild-config
node_modules/@expo/cli
  expo 50.0.0-alpha.0 - 50.0.19
  Depends on vulnerable versions of @expo/cli
node_modules/expo
  @expo/prebuild-config <=7.0.0-beta.0
  Depends on vulnerable versions of @expo/image-utils
node_modules/@expo/prebuild-config
  expo-splash-screen <=0.0.1-canary-20240109-93608d8 || 0.11.0 - 0.27.0
  Depends on vulnerable versions of @expo/prebuild-config
node_modules/expo-splash-screen
  expo-router <=0.0.33 || 2.0.0-rc.1 - 3.5.0
  Depends on vulnerable versions of expo-splash-screen
node_modules/expo-router
  expo-notifications <=0.0.1-canary-20240109-93608d8 || 0.9.0 - 0.27.8
  Depends on vulnerable versions of @expo/image-utils
node_modules/expo-notifications

```

tar <6.2.1

Vulnerabilidade: A biblioteca tar pode causar uma negação de serviço ao analisar um arquivo tar devido à falta de validação da contagem de pastas.

undici <=5.28.3

Vulnerabilidade: O undici tem problemas com o cabeçalho Proxy-Authorization em redirecionamentos cruzados e com a opção de integridade do fetch, onde o algoritmo é especificado, mas o valor do hash está incorreto.

```

tar <6.2.1
Severity: moderate
Denial of service while parsing a tar file due to lack of folders count validation - https://github.com/advisories/GHSA-fsx3-32g6-xq36
fix available via 'npm audit fix'
node_modules/tar

undici <=5.28.3
Undici's Proxy-Authorization header not cleared on cross-origin redirect for dispatch, request, stream, pipeline - https://github.com/advisories/GHSA-m4v8-wqvr-p9f7
Undici's fetch with integrity option is too lax when algorithm is specified but hash value is incorrect - https://github.com/advisories/GHSA-9qxr-qj54-h672
fix available via 'npm audit fix'
node_modules/undici
  @firebase/auth 1.5.0-20231112213030 - 1.7.0
  Depends on vulnerable versions of undici
node_modules/@firebase/auth
  firebase 0.90.22 || 7.9.1-0 - 7.9.1-canary.0396117e || 8.10.0-20217172214 - 8.10.0-canary.f40c0db53 || 9.0.0-20217250818 - 9.0.0-canary.d0d3acb10 || 9.0.1-2021727231341 - 9.0.1-canary.e039e1472 ||
  9.0.2-2021891633 - 9.0.2-canary.ff9baf70c || 9.0.3-202181503543 - 9.1.0-canary.f708324a1 || 9.1.1-2021830195733 - 9.1.1-canary.e70de6201 || 9.1.2-20219523556 - 9.1.2-canary.fc1d36497 || 9.19.1-202303
  31192943 - 9.19.1-canary.264909862 || 9.22.1-20230524195328 || 10.6.0-20231107192534 - 10.6.0-canary.ebc694a33 || 10.7.0-20231112213030 - 10.10.0 || 10.12.2-20240523105724 - 10.12.2-eap-firebase-backf
  iller.87bahf06c
  Depends on vulnerable versions of @firebase/auth
  Depends on vulnerable versions of @firebase/auth-compat
  Depends on vulnerable versions of @firebase/firestore
  Depends on vulnerable versions of @firebase/firestore-compat
  Depends on vulnerable versions of @firebase/functions
  Depends on vulnerable versions of @firebase/functions-compat
  Depends on vulnerable versions of @firebase/storage
  Depends on vulnerable versions of @firebase/storage-compat
node_modules/firebase
  @firebase/auth-compat 0.4.9-20231107192534 - 0.4.9-canary.a5c1a3594 || 0.5.0-20231112213030 - 0.5.5
  Depends on vulnerable versions of @firebase/auth
  Depends on vulnerable versions of undici
node_modules/@firebase/auth-compat
  @firebase/firestore 4.3.2-20231027003707 - 4.3.2-canary.ff1a6ec2c || 4.4.0-20231112213030 - 4.5.1
  Depends on vulnerable versions of undici
node_modules/@firebase/firestore
  @firebase/firestore-compat <=0.0.0-900-exp.520ca39d0 || 0.3.21-20231027003707 - 0.3.21-canary.ff1a6ec2c || 0.3.22-20231107192534 - 0.3.22-canary.ebc694a33 || 0.3.23-20231112213030 - 0.3.28 || 0.3.30
  -20240424141009 - 0.3.30-dataconnect-preview.f2ddc3d70
  Depends on vulnerable versions of @firebase/firestore
node_modules/@firebase/firestore-compat
  @firebase/functions 0.11.0-20231112213030 - 0.11.3
  Depends on vulnerable versions of undici
node_modules/@firebase/functions

```

ws 6.0.0 - 6.2.2 || 7.0.0 - 7.5.9 || 8.0.0 - 8.17.0

Vulnerabilidade: O ws é afetado por uma negação de serviço ao lidar com uma solicitação com muitos cabeçalhos HTTP

```
ws 6.0.0 - 6.2.2 || 7.0.0 - 7.5.9 || 8.0.0 - 8.17.0
Severity: high
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
fix available via `npm audit fix`
node_modules/@react-native-community/cli-server-api/node_modules/ws
node_modules/@react-native/dev-middleware/node_modules/ws
node_modules/metro/node_modules/ws
node_modules/react-devtools-core/node_modules/ws
node_modules/react-native/node_modules/ws
node_modules/ws
30 vulnerabilities (10 low, 11 moderate, 9 high)
```

CONSIDERAÇÕES FINAIS

A gestão da qualidade de software é essencial para garantir que os produtos digitais desenvolvidos atendam aos objetivos definidos pelo time de projeto. Ela envolve práticas de definição, gestão e melhoria dos processos, com foco nas atividades de levantamento e gestão de requisitos, verificação e validação (especialmente testes). Além disso, a qualidade de software não se resume apenas ao código bem escrito; também avalia fatores como estrutura, consistência, modularidade, legibilidade e manutenibilidade. Portanto, ao desenvolver software, é fundamental considerar a qualidade em todas as etapas do ciclo de vida do projeto.

Conclui-se que a qualidade de software é um aspecto crucial para garantir que os produtos de software atendam às necessidades dos usuários e funcionem conforme o esperado, uma documentação prévia e organizada facilita em grande escala a abordagem do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

Bartié, Alexandre Garantia da qualidade de software : adquirindo maturidade organizacional / Alexandre Bartié. – Rio de Janeiro : Elsevier, 2002 – 13a Reimpressão. ISBN 85-352-1124-1 1. Software – Testes. I. Título 02-1253.

Editora: Ciência Moderna, versão 2.0, ano 2013. VASQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. Engenharia de Requisitos: software orientado ao negócio. Software Orientado ao Negócio.

UML: Unified Modelling Language - Uma Ferramenta para o Design de Software.