INSTITUTO FEDERAL DE PERNAMBUCO

CAMPUS GARANHUNS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

SAMUEL DE SOUZA FERRAZ

ARRETADAS:
DESENVOLVIMENTO DE SOFTWARE DE PREVENÇÃO À VIOLÊNCIA CONTRA AS MULHERES

Garanhuns - PE

2024

SAMUEL DE SOUZA FERRAZ

ARRETADAS:
DESENVOLVIMENTO DE SOFTWARE DE PREVENÇÃO À VIOLÊNCIA CONTRA AS MULHERES

Trabalho de conclusão de curso apresentado ao Instituto Federal de Pernambuco Campus Garanhuns como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Alessandra Maranhão S. Sivini Siqueira

Garanhuns - PE

2024

F381a Ferraz, Samuel de Souza.
Arretadas : desenvolvimento de software de prevenção à violência contra as mulheres / Samuel de Souza Ferraz ; orientadora Alessandra Maranhão Soares Sivini Siqueira, 2024.
38f. : il.

Orientadora: Alessandra Maranhão Soares Sivini Siqueira.

Trabalho de Conclusão de Curso (Graduação) – Instituto Federal de Pernambuco. Pró-Reitoria de Ensino. Diretoria de Ensino. Campus Garanhuns. Coordenação do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, 2024.

1. Software - Desenvolvimento. 2. Software de aplicação – Desenvolvimento. 3. Violência contra as mulheres – Garanhuns (PE) – Prevenção – Software. 4. Violência familiar – Garanhuns (PE) – Prevenção – Software. I. Título.

CDD 005.1

Andréa Maria Lidington Lins –CRB4/868

SAMUEL DE SOUZA FERRAZ

ARRETADAS:

DESENVOLVIMENTO DE SOFTWARE DE PREVENÇÃO À VIOLÊNCIA CONTRA AS MULHERES

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Tecnólogo e aprovado em sua forma final pelo Curso de Análise e desenvolvimento de sistemas da Instituto Federal de Pernambuco Campus Garanhuns.

Garanhuns – PE, 05 de nov. 2024

gov.br Documento assinado digitalmente
Alessandra Maranhao Soares Sivini Siqueira
Data: 07/11/2024 21:47:22-0300
Verifique em https://validar.iti.gov.br

Me. Alessandra Maranhão Soares Sivini Siqueira (Orientadora)
Instituto Federal de Pernambuco Campus Garanhuns

gov.br Documento assinado digitalmente
GENESIS JEFERSON FERREIRA PEREIRA DE LIMA
Data: 08/11/2024 15:37:48-0300
Verifique em https://validar.iti.gov.br

Me. Gênesis Jeferson Ferreira Pereira de Lima
Instituto Federal de Pernambuco Campus Garanhuns

gov.br Documento assinado digitalmente
LEONARDO SOARES E SILVA
Data: 08/11/2024 08:59:53-0300
Verifique em https://validar.iti.gov.br

Me. Leonardo Soares e Silva
Instituto Federal de Pernambuco Campus Garanhuns

"Não aceito mais as coisas que não posso mudar, estou mudando as coisas que não posso aceitar."

ANGELA DAVIS

Dedico este trabalho a todos que apoiaram e tornaram este projeto possível a tempo.

AGRADECIMENTOS

Agradeço incomensuravelmente ao ilustríssimo Professor Dr. André Padilha por ter me fornecido esse modelo de trabalho em LaTeX e facilitar minha jornada na escrita acadêmica.

À minha família pelo incentivo e compreensão em momentos difíceis.

À Professora orientadora Me. Alessandra Siqueira pelo apoio durante o projeto de extensão.

Aos professores da instituição, em especial ao Professor Me. Leonardo Soares, pela excelente didática e pelo suporte recebido.

À coordenação do curso, em nome do Professor Dr. Rafael Mesquita, e à direção geral, representada pelo diretor Roberto Amaral, pela viabilidade de apresentar este projeto.

Por fim, agradeço a todos que contribuíram de alguma forma para a realização deste trabalho.

RESUMO

Conforme definido no artigo 1° da Convenção de Belém do Pará (1994), a violência contra a mulher é todo ato que ocasione um dano físico, sexual, psicológico e/ou patrimonial, motivado pela questão de gênero. Ademais, a violência contra a mulher em sua grande maioria ocorre no âmbito privado, ou seja, dentro do próprio lar, advindos de um parceiro ou familiar da vítima. Diversas iniciativas foram feitas para proteger as mulheres dessa violência, desde implementação e reavaliação de leis, delegacias da mulher, e atendimento psicossocial, a medidas de cunho tecnológico, facilitadas pela expansão do acesso à informação por meio de celulares conectados à internet. A utilização dos celulares como ferramenta de apoio no combate à violência motivou o desenvolvimento do sistema "Arretadas", cujo intuito é auxiliar vítimas nas comunidades da cidade de Garanhuns - PE, a partir de um aplicativo e página web, permitindo às vítimas as funcionalidades de pedido de socorro (similar ao botão de pânico, a ser utilizado em situações de risco iminente) e a emissão de denúncias anônimas, e às autoridades competentes o mapeamento e monitoramento de dados coletados por meio de gráficos, favorecendo a tomada de decisão quanto às políticas públicas eficientes.

Palavras-chave: Aplicativo; Desenvolvimento; Medidas; Mulher; Denúncias

ABSTRACT

As defined in article 1 of the Convention of Belém do Pará (1994), violence against women is any act that causes physical, sexual, psychological and property damage, motivated by gender. Furthermore, the vast majority of violence against women occurs in the private sphere, that is, within the home itself, originating from a partner or family member of the victim. Several initiatives were taken to protect women from this violence, from the implementation and reevaluation of laws, women's police stations, and psychosocial care, to technological measures, facilitated by the expansion of access to information through cell phones connected to the internet. The use of cell phones as a support tool in the fight against violence motivated the development of the "Arretadas" system, whose aim is to assist victims in communities in the city of Garanhuns - PE, using an application and web page, allowing victims the functions of request for help (similar to the panic button, to be used in situations of imminent risk) and the issuance of anonymous reports, and to the competent authorities the mapping and monitoring of data collected through graphs, favoring decision making regarding policies efficient public services.

Keywords: Application; Development; Measures; Women; Reports.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

INTRODUÇÃO

A violência sofrida contra a mulher pode afetar tanto sua saúde, quanto sua produtividade e habilidade de cuidar de si e da própria família. Quando repetitiva ou de alta intensidade, a agressão física pode provocar traumatismos e doenças crônicas, como possibilidade do desenvolvimento de hipertensão arterial, problemas gastrointestinais e transtornos mentais gerados pelos altos índices de estresse (LAWRENZ et al, 2018).

Ainda segundo Lawrenz et al. (2018), dentre os sintomas psicológicos que podem se manifestar nas vítimas estão negação, choque, confusão e medo. Além disso, a violência pode desencadear possíveis transtornos como depressão, ansiedade, transtorno de estresse pós-traumático (TEPT), ideação e tentativa de suicídio e abuso de substâncias.

Segundo estudos populacionais, a violência contra a mulher ocorre normalmente por parte de pessoas próximas às vítimas, dentre elas, parceiros e demais familiares. Nesse caso, pessoas estranhas representam estatisticamente um menor risco no que diz respeito à agressão.

Schraiber et al. (2002) aponta que cerca de 20% a 50% das mulheres ao redor do mundo já sofreram violência física na vida adulta ao menos uma vez na vida advinda de um parceiro. No Brasil, são as delegacias de defesa da mulher que fornecem os dados de denúncias nos principais estudos de caso. Estes também são centrados na violência doméstica: o parceiro é agressor em cerca de 77,6% dos casos registrados (SOARES, 1999). Recentemente, essa forma de violência vem sendo concebida como baseada em questões de gênero, saúde e direitos humanos (HEISE et al, 1999).

A tecnologia da informação, mediante a facilidade de acesso aos celulares e a crescente utilização de aplicativos por parte da população, pode ser uma importante aliada no combate à violência contra a mulher. No Brasil, em 2019, cerca de 230 milhões de smartphones estão em uso, e esse recuso favorece a aplicação em diversas áreas, desde a educação, saúde e cuidado integral (LOPES et al., 2019), além do próprio meio de prevenção à violência contra mulher.

Dessa forma, o presente trabalho objetiva desenvolver um sistema voltado para dispositivos móveis (ou seja, *smartphones*), com o intuito de informar e auxiliar mulheres em situação de risco e vítimas de violência nas comunidades de Garanhuns – PE, que, segundo informações da Secretaria da Mulher de Garanhuns, indicam que cerca de 600 mulheres possuem medida protetiva e mais 2.000 mulheres são, ou foram, acompanhadas pela secretaria no ano de 2019/2020.

O sistema pode ser dividido em duas visões bem definidas: a da mulher que deseja usar o aplicativo e a dos profissionais da Secretaria da Mulher.

Na perspectiva da mulher, usuária do aplicativo, o sistema proverá informações gerais dentro da temática da violência praticada contra a mulher para fins de conscientização, a saber: direitos reprodutivos e sexuais, informações sobre a Lei Maria da Penha, telefone de contato de órgãos públicos de apoio, entre outros. Além disso, o aplicativo também facilitará a emissão de denúncias e alertas de pedido de socorro, por meio do envio de mensagens ao contato telefônico de amigos de confiança previamente cadastrados ou para as autoridades competentes.

No segundo caso, sob o ponto de vista do profissional da Secretaria da Mulher, os dados coletados com o uso do aplicativo serão exibidos para as autoridades competentes no formato de gráficos em uma página web. Tal módulo favorece, então, uma visualização dos dados pelos órgãos competentes a fim de facilitar a tomada de decisão quanto a políticas públicas favoráveis à temática.

Denominado de “Arretadas”, o aplicativo será aplicado na cidade de Garanhuns, iniciando pela publicação, divulgação e orientação com pequenos grupos, como os de mulheres com medida protetiva, e posteriormente expandido de forma gradativa para toda comunidade assistida pelo município e cidades circunvizinhas.

1 DESENVOLVIMENTO DO APLICATIVO ARRETADAS

Este capítulo objetiva detalhar aspectos do desenvolvimento do aplicativo Arretadas.

1.1. Requisitos

Na **Tabela *1*** são indicados os requisitos funcionais do aplicativo Arretadas.

Tabela 1: Requisitos Funcionais

ID	Requisito	Descrição	Prioridade
RF-01	Enviar alerta	O sistema deve enviar alertas de pedido de socorro com a localização para amigos cadastrados.	Alta
RF-02	Registrar Denúncias	O sistema deve registrar denúncias, incluindo informações como a situação, local, data e hora do ocorrido. Esses dados devem ser armazenados e utilizados para análises posteriores	Alta
RF-03	Cadastrar Usuário	O sistema deve realizar o cadastro de usuários, solicitando perguntas de segurança e a escolha da cidade atual.	Alta
RF-04	Autenticar Usuário	O sistema deve possibilitar login seguro utilizando token para autenticação.	Alta
RF-05	Gerenciar contatos de confiança	O sistema deve gerenciar contatos de pessoas de confiança (adicionar/remover/editar) e verificar a lista antes de enviar pedidos de socorro.	Alta
RF-06	Exibir Informações	O sistema deve exibir informações importantes da Secretaria da Mulher sobre medidas de prevenção.	Média
RF-07	Exibir Agenda com Contatos Importantes	O sistema deve exibir agenda contatos importantes de acordo com a cidade cadastrada.	Média
RF-08	Trocar Senha	O sistema deve permitir que o usuário altere sua senha, utilizando perguntas de segurança para validação.	Média
RF-09	Recuperar Senha	O sistema deve permitir a recuperação	Média

		de senha por meio de perguntas de segurança.	
RF-10	Excluir Conta	O sistema deve oferecer a opção de exclusão da conta do usuário a qualquer momento.	Baixa
RF-11	Termos e Condições	O sistema deve exibir os termos e condições de uso, além das informações sobre a LGPD.	Baixa

Fonte: O Autor (2024)

Na **Tabela *2*** são indicados os requisitos não funcionais do aplicativo Arretadas.

Tabela 2: Requisitos Não-Funcionais

ID	Requisito	Descrição	Prioridade
RNF-01	Segurança	O cadastro de usuário exigirá uma confirmação para atender apenas as mulheres das cidades de Garanhuns e Monteiro.	Alta
RNF-02	Segurança	O botão de pedido de socorro enviará a localização, data e horário atual.	Alta
RNF-03	Segurança	O cadastro de amigos deverá aceitar até 5 amigos.	Alta
RNF-04	Usabilidade	O botão de pedido de socorro e o botão de denúncia deverão estar em local de fácil acesso e de forma intuitiva.	Alta
RNF-05	Segurança	O botão de pedido de socorro enviará alertas para o WhatsApp das autoridades de segurança somente se a usuária tiver uma medida protetiva ativa, mantendo sua privacidade.	Alta
RNF-06	Desempenho	A resposta da requisição deve ser fornecida no máximo em 10 segundos.	Média

RNF-07	Conformidade Legal	O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados (LGPD), garantindo a proteção e privacidade dos dados pessoais das usuárias.	Alta

Fonte: O Autor (2024)

1.2. Sobre o Sistema "Arretadas"

O sistema "Arretadas" começou a ser desenvolvido em 2020 e atualmente tem como foco a sua disponibilização ao público-alvo, com acompanhamento para manutenção, melhorias de requisitos, testes e aprofundamento quanto à segurança de dados.

A priori, o sistema foi subdividido em três partes bem definidas: o aplicativo *mobile*, a página web de relatórios e o servidor.

Para desenvolvimento do aplicativo móvel, a tecnologia utilizada foi o *framework Flutter*, que é um framework *open source* desenvolvido pelo Google, que facilita o desenvolvimento de código nativo para múltiplas plataformas (como Android e iOS, por exemplo) a partir de um código-fonte único. Além disso, é produtivo ao permitir a execução do aplicativo em um *smartphone* ou emulador durante a programação.

Para o desenvolvimento do módulo de relatórios (*página web*) foi utilizado o *framework Javascript Vue.Js*, útil para a construção de interface gráfica favorável a uma maior interação e experiência do usuário. Este módulo utiliza os dados coletados com as denúncias e pedidos de socorro para gerar gráficos favoráveis à análise pelos órgãos competentes. Dessa forma, é possível que, a partir dessas informações, a tomada de decisão quanto a medidas públicas de combate a esses casos de violência seja facilitada.

No que diz respeito ao servidor, a API (*Application Programming Interface*) foi desenvolvida em *NodeJS*[1] e o banco de dados aplicado foi o *MongoDB*, o qual é orientado a documentos, livre e de código aberto.

A ferramenta *Postman* foi utilizada para executar testes de requisição na API, sendo possível um desenvolvimento mais eficiente na construção de requisições e análise das respostas

[1] Node.JS é um ambiente de execução de JavaScript no servidor, utilizado para criar aplicações rápidas e escaláveis, especialmente em tempo real, graças à sua arquitetura assíncrona.

enviadas pela API. O *deploy* e hospedagem da API estão no *Heroku*, plataforma na nuvem que suporta diversas tecnologias de programação.

Para controle de versão de código, auxiliando no controle de novas funcionalidades e compartilhamento entre desenvolvedores, análise e resolução de conflitos, utilizou-se o *Git* e o *GitHub*. Para edição do código-fonte, foi utilizada a ferramenta *Visual Studio Code*. Para testar o aplicativo em diversos dispositivos virtuais, o emulador do *Android Studio* foi utilizado.

Quanto aos recursos físicos demandados até então para o desenvolvimento do sistema, foi necessário basicamente os *smartphones* e computadores pessoais e os disponíveis na instituição, com os *softwares* elencados funcionando.

Quanto à gerência do projeto, foram realizadas reuniões semanais entre a equipe de desenvolvimento para acompanhamento das atividades, seguindo uma adaptação da metodologia ágil SCRUM.

Conforme bem apresenta seus idealizadores Schwaber e Beedle (2002), nas reuniões foram identificadas as atividades a fazer (*to do*), em execução (*doing*), e concluídas (*done*), acompanhadas por meio de um quadro de atividades no *Trello*[2]. Além disso, nas reuniões são pontuados impedimentos e dificuldades, a fim de serem solucionados em tempo hábil para evitar atrasos.

O projeto atualmente pode ser organizado nas seguintes etapas:

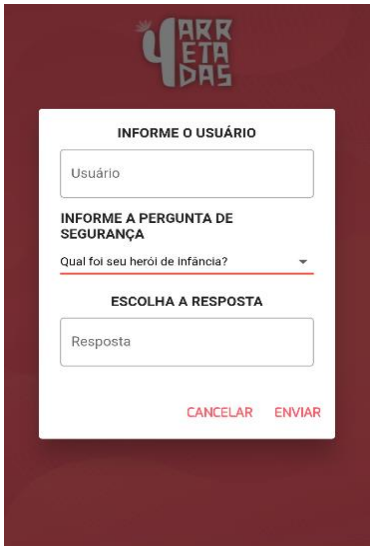
1) Manutenção e testes internos: Nesta etapa, os módulos principais do sistema "Arretadas" são aprimorados quanto a melhorias levantadas do ponto de vista interno (arquitetura do sistema) e externo (funcionalidades).
2) Publicação e validação dos usuários finais;
3) Divulgação do aplicativo e participação em eventos.

As Figuras 1, 2 e 3 exemplificam as telas do aplicativo "Arretadas".

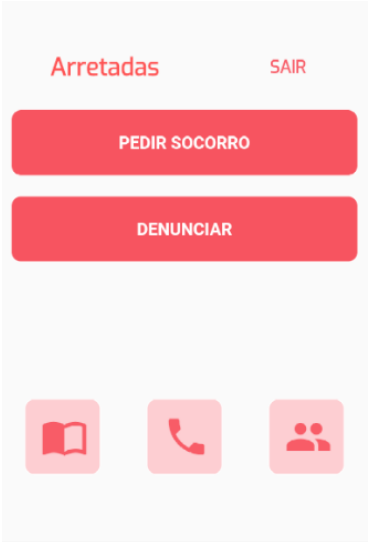
[2] *Trello* é uma ferramenta de gerenciamento de projetos que organiza tarefas em quadros, listas e cartões, permitindo a colaboração em tempo real e o acompanhamento do progresso das atividades.

Figura 1: Telas do Aplicativo “Arretadas” – Parte 1

Página de Login



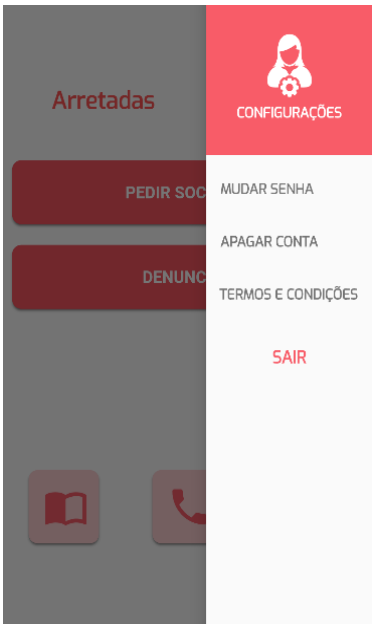
Página de Recuperação de Senha



Página Inicial do Aplicativo

Fonte: O Autor (2024)

Figura 2: Telas do Aplicativo “Arretadas” – Parte 2



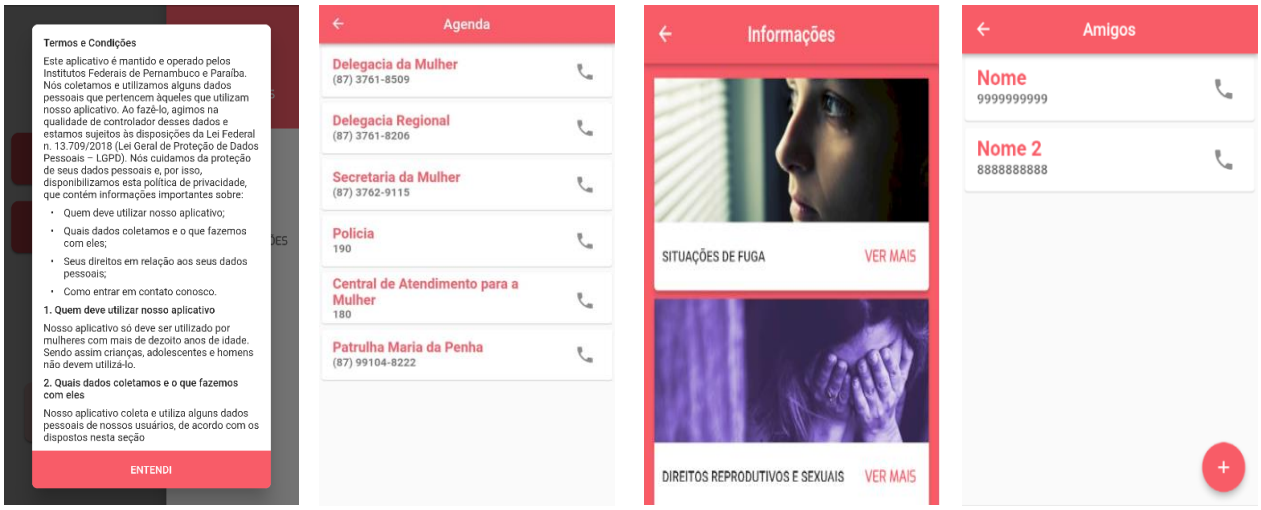
Navegação Lateral

Página de Pedido de Socorro

Formulário para Denúncias

Fonte: O Autor (2024)

Figura 3: Telas do Aplicativo “Arretadas” – Parte 3



Termos & Condições | Agenda com Contatos Importantes | Página Informativa | Página para Registro de Contatos Amigos

Fonte: O Autor (2024)

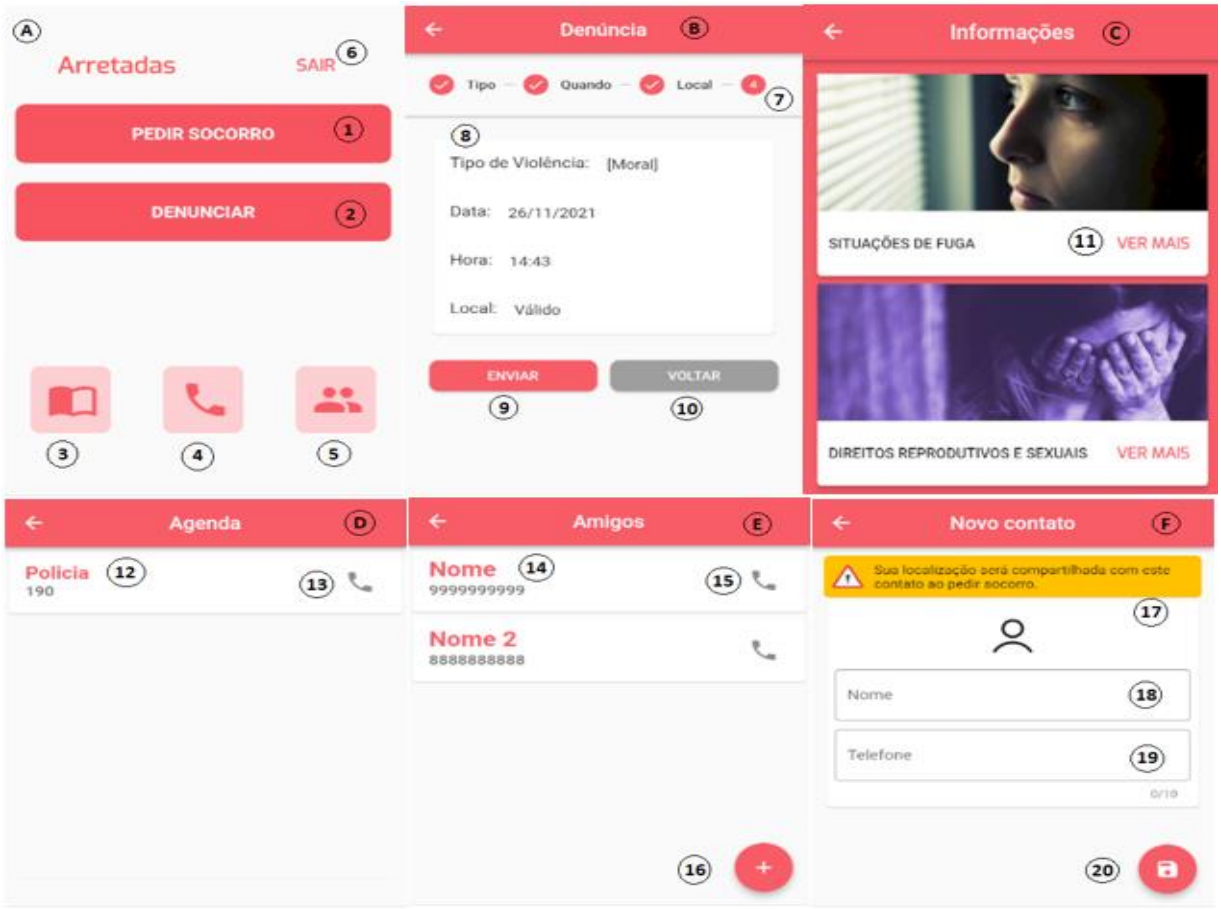
Dentre as principais funcionalidades (Figura 4) desenvolvidas estão:

1) Cadastro e Autenticação (*login*) da usuária
   a) Para cadastro, a mulher precisa somente registrar um *nickname*, uma senha, o código de medida protetiva (opcional), escolher uma pergunta secreta e respondê-la.
   b) Não é registrado email para fins de evitar dispor de dados sensíveis das usuárias, por isso, para fins de recuperação de senha, foi utilizada a técnica da pergunta secreta, de modo que, quando a mulher precisar recuperar a sua senha, deverá responder a essa pergunta corretamente.
   c) O código de medida protetiva, por sua vez, será gerado por um administrador da Secretaria da Mulher, a fim de disponibilizar um *voucher* apenas às usuárias que possuem tal medida protetiva (isto é, mecanismo legal que tem o objetivo de proteger um indivíduo que esteja em situação de risco). Um programa *desktop* básico (que pode ser executado em qualquer computador), escrito em linguagem *Java*, foi desenvolvido para gerar esse código e enviá-lo ao banco de dados do “Arretadas”. A secretaria deverá entregar o código para uma usuária, que irá usá-lo para efetivar seu cadastro no aplicativo, e assim, a mulher que possui condição

especial de medida protetiva poderá ter sua localização enviada à polícia quando solicitar socorro.

2) Pedir Socorro (Figura 7, Tela A)
    a) A função envia uma mensagem padrão para o Whatsapp de até 5 contatos cadastrados como amigos de confiança. No caso das mulheres com medida protetiva, também será enviada a mensagem ao contato da polícia. Essa mensagem automática possui um *link* com a localização da mulher captada por meio do GPS do *smartphone*. Este link poderá ser aberto no aplicativo Google Maps para ajudar na localização da pessoa que solicitou socorro.
3) Denunciar (Figura 7, Tela B)
    a) Para fins de registrar uma situação de violência sofrida, a mulher pode submeter sua denúncia por meio do aplicativo, indicando o tipo de violência sofrida, data, hora e local do ocorrido (opcional).
    b) Diferentemente do Pedir Socorro, a denúncia pode ser feita em um momento sem caráter emergencial, posterior à violência sofrida.
4) Informações (Figura 7, Tela C)
    a) Permite que a mulher leia uma série de artigos selecionados pela Secretaria da Mulher sobre a temática da violência.
5) Contatos Úteis (Figura 7, Tela D)
    a) Lista alguns contatos principais de autoridades competentes, como o da Polícia e o da Secretaria da Mulher.
6) Amigos (Figura 7, Telas E e F)
    a) Cadastra e lista os contatos de pessoas de confiança da mulher, a fim de que eles recebam pelo Whatsapp uma mensagem automática com a localização da mulher, caso ela solicite socorro pelo aplicativo.

Figura 4: Funcionalidades Aplicativo “Arretadas”



Fonte: O Autor (2024)

A **Tabela *3*** identifica as funcionalidades ilustradas nas telas anteriores.

Tabela 3: Registros de Funcionalidades Aplicativo “Arretadas”

A	1	Pedir Socorro	C	11	Informações Gerais
A	2	Denunciar	D	12	Detalhes dos Contatos Úteis (Nome e Telefone)
A	3	Informações	D	13	Ligar para contato da Agenda
A	4	Contatos Úteis	E	14	Detalhes dos Contatos Amigos (Nome e Telefone)
A	5	Amigos	E	15	Ligar para contato amigo
A	6	Sair do Aplicativo	E	16	Adicionar novo amigo
B	7	Barra de Progresso do Preenchimento do Formulário de Denúncia	F	17	Informação de Compartilhamento de Localização com contato cadastrado
B	8	Resumo das Informações da Denúncia	F	18	Nome do Amigo a ser cadastrado
B	9	Enviar Denúncia	F	19	Telefone do Amigo a ser cadastrado
B	10	Retroceder no formulário de Denúncia	F	20	Salvar novo amigo

Fonte: O Autor (2024)

Quanto às etapas do projeto, registram-se os seguintes resultados:

1) Manutenção e testes internos:
   a) No aplicativo, foram realizadas a manutenção do cadastro e mensagens de erro, implementação da funcionalidade de recuperação de senha com perguntas de segurança, filtragem de contatos importantes por cidade, ajustes na interface e finalização da funcionalidade de pedido de socorro (com o desenvolvimento e integração de um *bot* para conexão com o aplicativo *Whatsapp*);
   b) No módulo de relatórios, foram realizados ajustes mais pontuais, como melhorias nas mensagens de alerta, responsividade do site e redirecionamento para página de erro e login;
   c) Foi desenvolvido um estudo quanto à segurança dos dados para fins de adaptação à Lei Geral de Proteção de Dados (LGPD). Quanto a esse ponto, o sistema “Arretadas” evita a coleta de dados pessoais sensíveis dos usuários, tais como nome, endereço, CPF, etc. Além disso, adicionou-se ao sistema, os “termos e condições” para ajustar-se à LGPD;
2) Publicação e validação dos usuários finais:

a) Ambos módulos foram migrados para *deploy* e hospedagem na *Google Cloud Platform*, com o objetivo de viabilizar a disponibilidade do sistema (anteriormente disponibilizado apelas no *Heroku*);
b) Vale ressaltar que os recursos financeiros necessários para possibilitar a publicação do sistema, foram adquiridos em parceria com o professor Tiago Brasileiro, colaborador externo e professor do Instituto Federal da Paraíba – Campus Monteiro;
c) O aplicativo foi disponibilizado na *Play Store* e encontra-se em fase de testes alfa, restrito à equipe de desenvolvimento. Ressalta-se que essa disponibilização precisa ser feita de forma gradativa, inicialmente para um público restrito da Secretaria da Mulher, posteriormente para apenas mulheres com medida protetiva e, por fim, ao público-alvo geral da região.

3) Divulgação do aplicativo e participação em eventos
a) O projeto foi apresentado e divulgado no IX Encontro de Extensão do IFPE. Além disso, nesta apresentação as funcionalidades foram demonstradas como fins de tutorial para os usuários.

1.3. Design do Software

> "A Clean Architecture, proposta por Robert C. Martin, é uma abordagem de design de software que visa criar sistemas com baixa dependência entre módulos, facilitando a manutenção e a evolução do código ao longo do tempo. Esse modelo organiza o software em camadas, onde cada uma possui responsabilidades específicas e se comunica apenas com as camadas adjacentes, promovendo a separação de interesses e aumentando a testabilidade e reutilização do código. (MARTIN, 2020)"

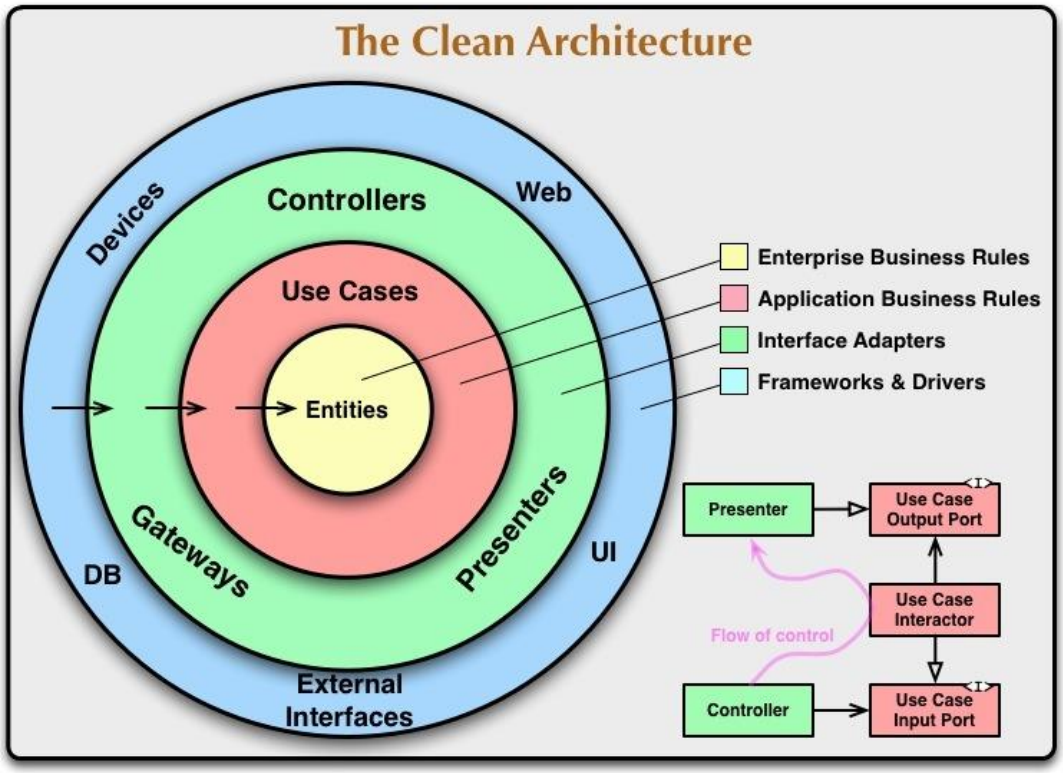
A Clean Architecture oferece um design modular que prioriza a independência entre as partes do sistema, separando o código em camadas para garantir que alterações em uma camada afetem o mínimo possível outras partes do software.

Essa arquitetura é composta por quatro principais camadas: Entidades, Casos de Uso, Adaptadores de Interface e Frameworks e Drivers. Cada camada possui responsabilidades específicas e segue o princípio da dependência interna, onde as camadas mais externas dependem das camadas mais internas, mas nunca o contrário. Isso permite que as camadas internas, que

geralmente contêm a lógica de negócios, permaneçam livres de dependências diretas com tecnologias externas, tornando o sistema mais flexível e adaptável a mudanças.

A Figura 1 ilustra a estrutura da Clean Architecture, que se baseia na orientação das dependências de fora para dentro, onde as camadas internas não têm conhecimento das externas. Na prática, essa abordagem permite que o núcleo do sistema seja independente de detalhes específicos de implementação, como bancos de dados, frameworks e interfaces de usuário. A camada de Entidades contém as regras mais gerais, enquanto a camada de Casos de Uso especifica os processos centrais da aplicação. Já a camada de Adaptadores de Interface cuida da conversão entre o domínio e a interface do usuário, enquanto a camada mais externa, Frameworks e Drivers, integra componentes externos ao sistema.

Figura 5: Diagrama da Estrutura da Clean Architecture



Fonte: Uncle Bob (2019)

1.3.1. Proposta de Adaptação para o *Flutter* com *Clean Dart*

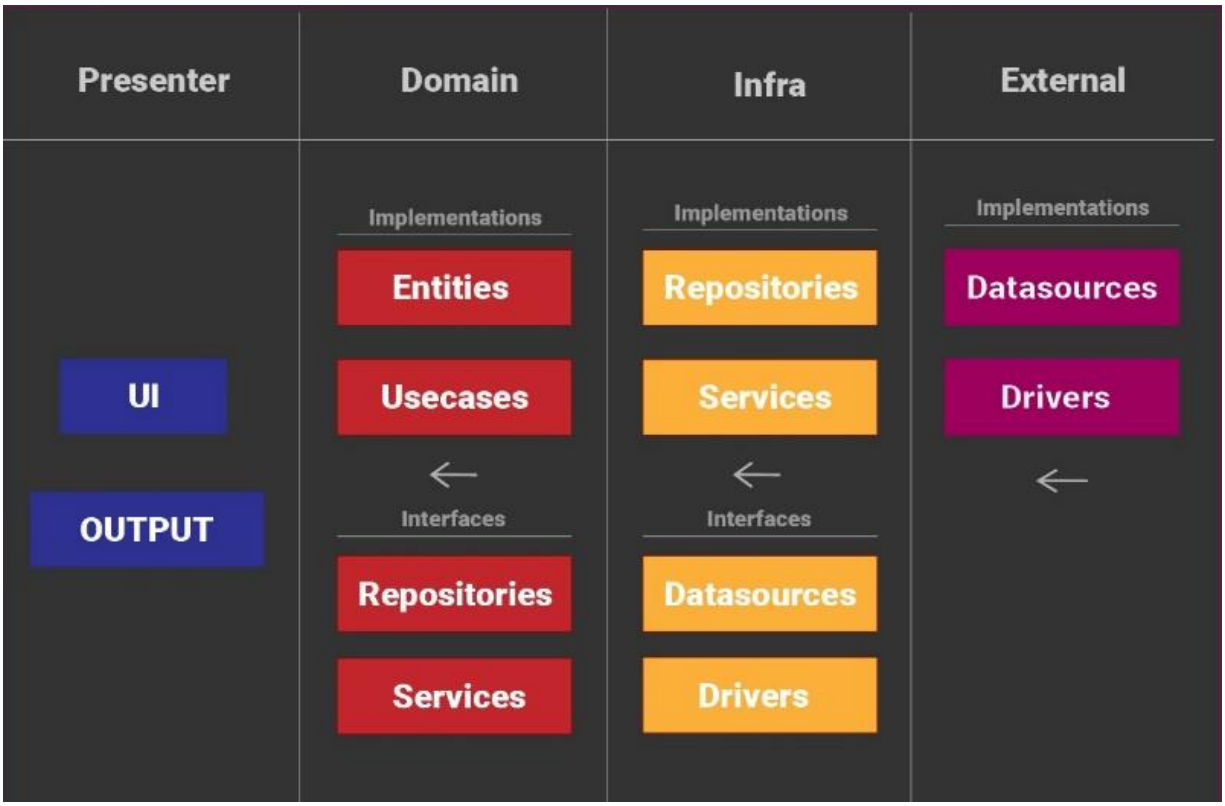
O Clean Dart é uma arquitetura voltada para o ecossistema Flutter, baseada na Clean Architecture, mas adaptada para as necessidades específicas da plataforma. Enquanto a Clean Architecture tradicional é uma abordagem genérica aplicável a várias plataformas, o Clean Dart ajusta esses princípios para atender a características essenciais do desenvolvimento com Flutter, como a gestão de estado e a integração com APIs externas. Essa estrutura modular organiza o projeto em camadas independentes, promovendo eficiência, produtividade e uma separação clara de responsabilidades. (FLUTTERANDO, 2020). As camadas incluem:

- ***Domain***: a camada de *Domain* é o núcleo da aplicação e representa conceitos do mundo real em termos computacionais. Ela é composta por Entidades (*Entities*), Invariantes do Domínio (*Errors*), Casos de Uso (*UseCases*) e Interfaces de Repositórios (*Repositories Interfaces*). As Entidades cuidam das regras de negócio no nível do domínio, enquanto os Casos de Uso controlam as regras de negócio no nível da aplicação, acionando os contratos necessários. As Interfaces de Repositórios definem os contratos para o acesso aos dados, garantindo que o domínio permaneça independente da camada de persistência.
- ***Infra***: implementa as interfaces dos repositórios definidos na camada de domínio. Contém a pasta *Repositories* com as implementações dos repositórios, e *Datasources,* onde estão as interfaces dos *datasources* que abstraem a fonte de dados (API, banco de dados local, etc.).
- ***External****:* lida com a implementação dos *Datasources* para o consumo de APIs, banco de dados ou outras fontes de dados externas. Por exemplo, ao consumir uma API REST, a implementação específica do *datasource* estará nesta camada.
- ***Presenter***: camada que cuida da *interface* com o usuário. Contém os Stores (gerenciadores de estado), responsáveis por consumir os *UseCases* e gerenciar o estado da aplicação. Além disso, abriga as *Pages* que se conectam aos *Stores*, formando a interface visual da aplicação.

Na Figura 2, vemos uma adaptação detalhada da *Clean Architecture* para o *Flutter* com *Clean Dart*. A camada *Domain* define as regras de negócio e abstrações do sistema, e seus *UseCases* são consumidos pelos *Stores* na camada de *Presenter*, que lida com a lógica de estado e apresentação. As *Pages* interagem diretamente com os *Stores*, refletindo mudanças de estado na

interface de usuário. A camada *Infra*, por sua vez, implementa as interfaces de repositórios e *datasources* definidos no Domain, mantendo-se independente das especificidades da *External*, que lida com o consumo direto de APIs ou banco de dados.

Figura 6: Estrutura da Clean Architecture adaptada para Flutter com Clean Dart



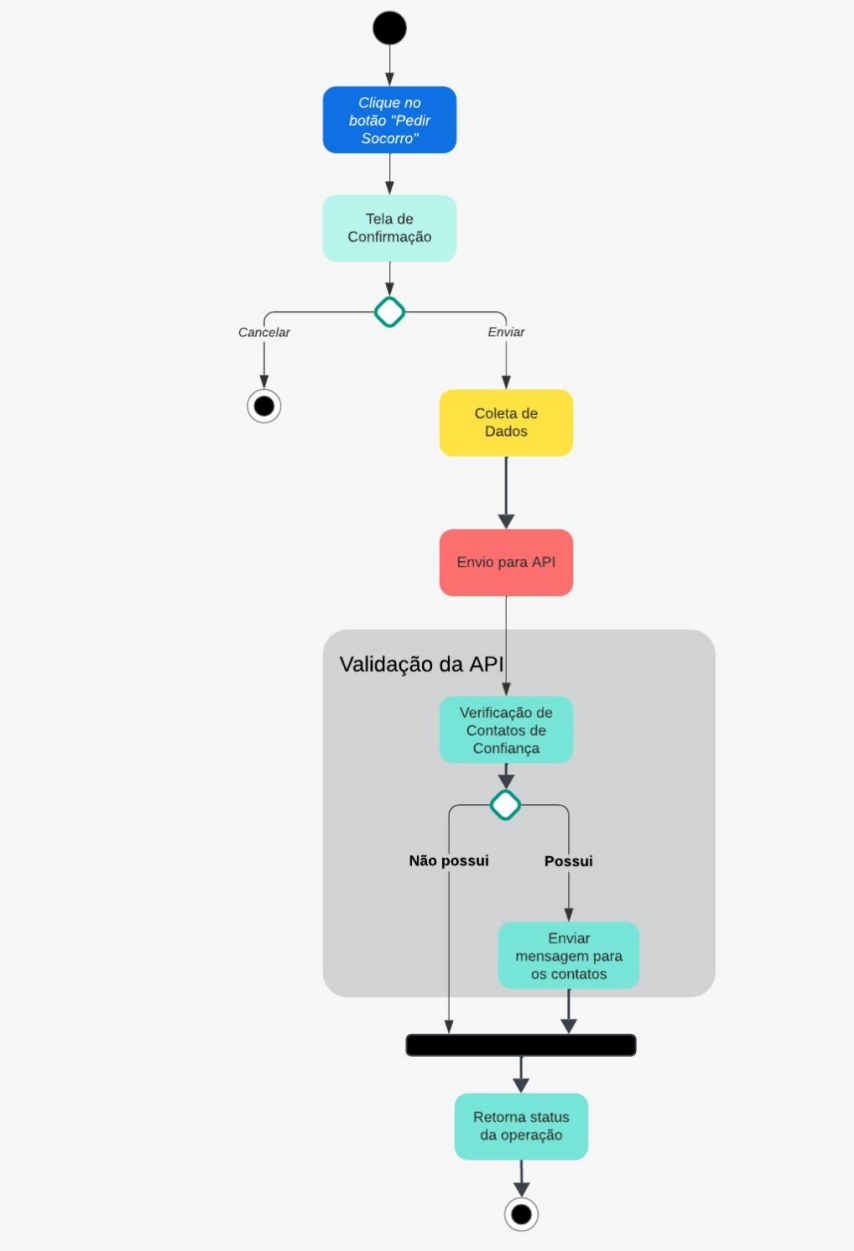
Fonte: O Autor (2024)

Essa organização permite isolar a lógica de negócios e o estado da aplicação do restante do código, promovendo uma estrutura flexível e testável. Adaptar a *Clean Architecture* ao *Clean Dart* fortalece a modularidade, a escalabilidade e a manutenibilidade da aplicação no contexto *Flutter*.

1.4. Diagrama de Atividades

O fluxograma detalha o processo da funcionalidade de Pedir Socorro dentro da aplicação, que permite ao usuário solicitar ajuda em situações de emergência com base na sua localização, data e hora atual. Esta funcionalidade utiliza informações pré-cadastradas para direcionar o pedido de socorro ao destinatário apropriado (autoridades ou contatos de confiança).

Figura 7: Diagrama de Atividades - Fluxo de Pedir Socorro



Fonte: O Autor (2024)

2 CASOS DE TESTES E RELATÓRIO DE EXECUÇÃO DE TESTES

2.1. Estrutura de Testes

O desenvolvimento do sistema **Arretadas** envolveu uma série de testes para garantir que as funcionalidades do aplicativo estivessem operando corretamente antes do lançamento.

Esses testes foram realizados pela equipe de desenvolvimento com o objetivo de validar se o sistema atendia aos requisitos funcionais e de usabilidade. A estrutura de testes adotada envolveu principalmente testes manuais, considerando o estágio de desenvolvimento e a natureza do sistema.

Abaixo está a organização dos testes realizados:

- **Testes Funcionais**: Validaram se as funcionalidades essenciais, como o envio de alertas e a realização de denúncias, estavam funcionando conforme o esperado.
- **Testes de Usabilidade**: Avaliaram a interação do usuário com a interface, buscando garantir uma navegação intuitiva e fluida.
- **Testes de Integração**: Verificaram a integração entre aplicativo móvel, servidor e página web de relatórios, garantindo que os dados fossem trocados corretamente entre plataformas.
- **Testes de Desempenho**: Embora não tenham sido realizados testes de estresse, foram verificados aspectos de desempenho durante o uso normal do sistema para garantir uma experiência sem falhas.

2.2. Casos de Teste

A seguir estão os casos de teste para as principais funcionalidades do sistema **Arretadas**:

1) **Pedir Socorro**
    - **Objetivo**: Verificar se a funcionalidade de pedir socorro funciona corretamente.
    - **Passos**:
        - Abrir o aplicativo, realizar o login e clicar no botão “Pedir Socorro” na tela de Menu Principal.
        - Confirmar se deseja realmente pedir socorro.
        - Acionar o botão "Enviar".
    - **Resultado Esperado**: A localização é capturada e o pedido de socorro é enviado para os contatos pré-cadastrados.

- **Status**: Passou

2) **Denunciar**
   - **Objetivo**: Validar o envio correto da denúncia.
   - **Passos**:
     - Acessar a tela de denúncia e preencher o formulário.
     - Enviar a denúncia.
   - **Resultado Esperado**: Denúncia enviada com sucesso e confirmação exibida.
   - **Status**: Passou
3) **Adicionar Novo Amigo**
   - **Objetivo**: Validar a adição de um novo amigo à lista.
   - **Passos**:
     - Acessar a seção "Amigos", adicionar um novo amigo e salvar.
   - **Resultado Esperado**: O novo amigo é adicionado corretamente à lista.
   - **Status**: Passou

2.3. Plano de Testes

O plano de testes foi estruturado da seguinte forma:

- **Escopo**: O teste abrangeu todas as funcionalidades principais do aplicativo **Arretadas**, incluindo pedidos de socorro, denúncias, e o gerenciamento de amigos.
- **Recursos**: A equipe de desenvolvimento foi responsável pela execução dos testes, e foram utilizados emuladores e dispositivos reais para verificar o comportamento do aplicativo.
- **Métodos de Teste**: Os testes foram realizados manualmente pela equipe de desenvolvimento, com uma abordagem exploratória para cobrir o máximo de cenários.
- **Critérios de Aceitação**: Para que uma funcionalidade fosse considerada aprovada, era necessário que o fluxo completo fosse executado sem falhas críticas, e que o sistema respondesse adequadamente às ações do usuário.

2.4. Resultados e Análises

Durante os testes de aceitação, a equipe encontrou alguns pequenos ajustes a serem feitos em relação ao desempenho e à interface do usuário, mas não foram identificados bugs críticos. As

funcionalidades principais, como o envio de pedidos de socorro e denúncias, funcionaram conforme o esperado. A seguir, estão as análises dos testes realizados:

- **Pedir Socorro**: A funcionalidade de pedir socorro foi testada com sucesso em múltiplos cenários, incluindo a verificação da localização e o teste sem conexão com a internet. Não foram observados problemas durante a execução do teste.
- **Denúncias**: A função de envio de denúncias também foi validada com êxito. Após o envio, o aplicativo exibia uma mensagem de confirmação, garantindo que a denúncia foi registrada corretamente.
- **Adição de Amigos**: O processo de adicionar amigos foi validado com sucesso, e os contatos ficaram salvos na lista do aplicativo.

Em relação às melhorias futuras, recomenda-se a implementação de **testes automatizados** para otimizar o processo de verificação contínua, especialmente para as funcionalidades críticas. Também foi sugerido o desenvolvimento de testes de **segurança** para garantir a proteção dos dados sensíveis, como as informações pessoais dos usuários.

3 CONCLUSÃO

Por fim, conclui-se que, apesar do aplicativo não ter sido implantado e disponibilizado ao público-alvo oficialmente, no último ano de desenvolvimento houve um amadurecimento quanto à arquitetura do sistema, a fim de facilitar a sua manutenção posterior, somado a diversas melhorias no design, desenvolvimento de módulo de relatórios, expansão de funcionalidades, testes, documentação e aprofundamento nas questões de segurança, relevantes para que o sistema esteja alinhado com a Lei Geral de Proteção de Dados, e publicação na loja de aplicativos.

Ademais, o sistema deverá ser um potencial divisor de águas no quesito de combate à violência contra a mulher na região de Garanhuns. Destaca-se ainda que além do IFPE, o IFPB (Campus Monteiro) também atuou como parceiro no projeto, por meio do docente Tiago Brasileiro Araújo e de estudantes da mesma instituição, contribuindo no desenvolvimento de algumas funcionalidades, dentre elas, a integração com o *Whatsapp* por meio de um *Chatbot*.

REFERÊNCIAS

CONVENÇÃO DE BELÉM DO PARÁ – **Convenção Interamericana para prevenir, punir e erradicar a violência contra a mulher**, 1994.

HEISE, L.; ELLSBERG, M.; GOTTEMOELLER, M. ***Ending Violence against women***. Popul Rep, 27(4):1-43, 1999.

LAWRENZ, P. et al. **Violência contra Mulher: Notificações dos Profissionais da Saúde no Rio Grande do Sul**. Psicologia: Teoria e Pesquisa, v. 34, 2018.

LOPES, J. P. et al. **Inovações tecnológicas para dispositivos móveis no cuidado em vacinação. Journal of Health Informatics**, v. 11, n. 2, 2019.

MARTIN, R. C. **Arquitetura Limpa: O Guia do Artesão para Estrutura e Design de Software.** Rio de Janeiro, RJ: Alta Books, 2020.

SCHRAIBER, L. B. **Violência contra a mulher: estudo em uma unidade de atenção primária à saúde**. Revista de Saúde Pública, v. 36, n. 4, p. 470-477, 2002.

SCHWABER, K.; BEEDLE, M. **Agile software development with Scrum**. Upper Saddle River: Prentice Hall, 2002.

SOARES, B.S. **Mulheres Invisíveis: Violência Conjugal e Novas Políticas de Segurança.** Rio de Janeiro: Civilização Brasileira, 1999.

FLUTTERANDO. **Clean Dart - Proposta de Arquitetura Limpa para o Dart/Flutter.** GitHub, 2020**. Disponível em:** https://github.com/Flutterando/Clean-Dart. Acesso em: 05 nov. 2024.

APÊNDICES

Apêndice 1: Cronograma de Atividades

Fonte: O Autor (2024)

Apêndice 2: Camada de Apresentação - Implementação do Store para gerenciar o estado do envio de alerta

Exemplo 1.1: Camada de apresentação - Implementação do store para gerenciar o estado do envio de alerta

```
1 class AlertStore extends NotifierStore<AlertException, String> {
2   final IAlertUsecase alertUsecase;
3
4   AlertStore(this.alertUsecase) : super('');
5
6   void sendAlert(AlertParams params) {
7     executeEither(() =>
     EitherAdapterImpl.adapter(alertUsecase(params)));
8   }
9 }
```

Fonte: O Autor (2024)

Apêndice 3: Camada de Domínio – Implementação do caso de uso Emitir Alerta

Exemplo 1.2: Camada de domínio - Implementação do caso de uso Emitir Alerta

```
class AlertUsecase implements IAlertUsecase {
  final AlertRepository repository;

  AlertUsecase(this.repository);

  @override
  Future<Either<AlertException, String>> call(AlertParams
   params) async {
    if (params.userId.isEmpty) {
      return Left(AlertException('Usuário inválido'));
    }
    if (params.latitude.isEmpty) {
      return Left(AlertException('Latitude inválida'));
    }
    if (params.longitude.isEmpty) {
      return Left(AlertException('Longitude inválida'));
    }
    if (params.date.isEmpty) {
      return Left(AlertException('Data inválida'));
    }
    if (params.hour.isEmpty) {
      return Left(AlertException('Hora inválida'));
    }
    return await repository.sendAlert(params);
  }
}
```

Fonte: O Autor (2024)

Apêndice 4: Camada de Apresentação – Implementação do Repository que chama o datasource

Exemplo 1.3: Camada de infraestrutura - Implementação do repository que chama o datasource

```
class AlertRepositoryImpl implements AlertRepository {
  final AlertDatasource datasource;

  AlertRepositoryImpl(this.datasource);

  @override
  Future<Either<AlertException, String>> sendAlert(AlertParams
   params) async {
    try {
      final response = await datasource.sendAlert(params);
      return Right(response);
    } on AlertException catch (e) {
      return Left(e);
    }
  }
}
```

Fonte: O Autor (2024)

Apêndice 5: Camada Externa – Implementação do DataSource para enviar alerta para API

Exemplo 1.4: Camada externa - Implementação do datasource para enviar alerta para Api

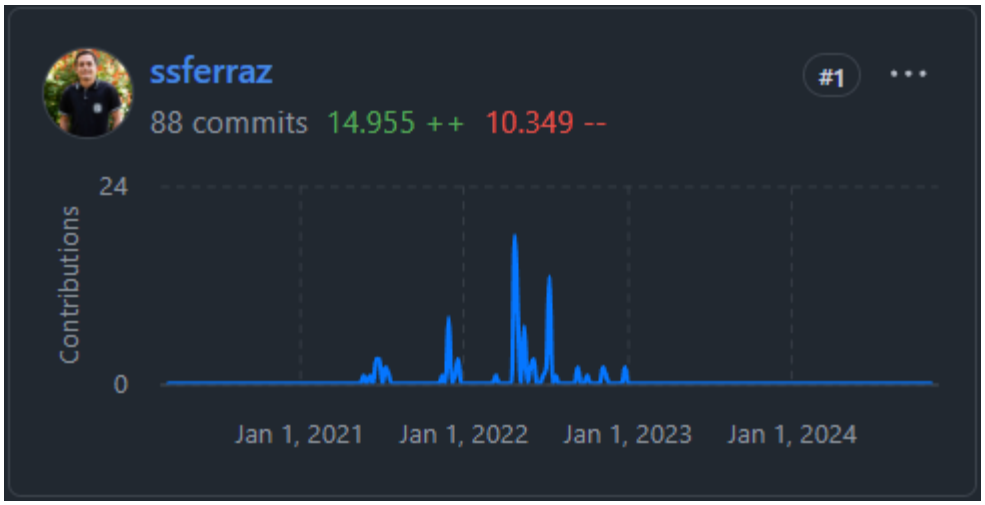
```
class AlertApi implements AlertDatasource {
  final Dio dio;

  AlertApi(this.dio);

  @override
  Future<String> sendAlert(AlertParams params) async {
    dio.options.connectTimeout = 10000;
    dio.options.headers['authorization'] = 'Bearer
   ${params.token}';
    try {
      await dio.post('${ApiEndpoint.urlProducao}/alert', data: {
        'id': params.userId,
        'latitude': params.latitude,
        'longitude': params.longitude,
        'date': params.date,
        'hour': params.hour,
      });
      return "Pedido de socorro enviado com sucesso!";
    } on DioError catch (e) {
      if (e.type == DioErrorType.connectTimeout ||
          e.type == DioErrorType.receiveTimeout) {
        throw AlertException(
            "Servidor FORA DO AR! Tente novamente mais tarde!");
      } else if (e.type == DioErrorType.other) {
        throw AlertException("Sem conexão com a Internet");
      } else {
        throw AlertException(e.message);
      }
    }
  }
}
```

Fonte: O Autor (2024)

Apêndice 6: Registro do Controle de Versão



https://github.com/DataLabIFPE/arretadas

Fonte: O Autor (2024)

Apêndice 7: Link do Aplicativo publicado na PlayStore

https://play.google.com/store/apps/details?id=com.github.datalab.ifpe.arretadas

Fonte: O Autor (2024)