

Projeto Interdisciplinar I
2^a Etapa – Projeto da Solução

G.U.I.A
GESTÃO UNIFICADA DE INFORMAÇÕES DO
AGENTE

Grupo J

Integrantes do Grupo:

Alessandro Sondey Rodrigues Lima
Milena Pianaro Caetano

Orientador(a): Eliana Santos

Co-orientador(a): Lauriana Paludo

Pinhais
2025

1. Introdução

O presente projeto propõe o desenvolvimento do aplicativo G.U.I.A — Gestão Unificada de Informação e Acompanhamento — como uma ferramenta tecnológica voltada à modernização das atividades dos ACS. O sistema será concebido para centralizar informações, automatizar tarefas recorrentes, otimizar deslocamentos por meio de geolocalização e roteirização inteligente, e fortalecer a comunicação entre equipes de saúde e comunidade. Além disso, será projetado para funcionar em modo offline, garantindo acessibilidade mesmo em áreas com baixa ou nenhuma conectividade.

Ao integrar funcionalidades como registro digital de visitas, alertas personalizados, mapas de áreas de risco e canais de comunicação direta, o G.U.I.A busca não apenas apoiar a rotina operacional dos ACS, mas também contribuir para o monitoramento estratégico por parte dos gestores. Alinhado às diretrizes de transformação digital do SUS e aos Objetivos de Desenvolvimento Sustentável (ODS), especialmente o ODS 3 — Saúde e Bem-Estar, o aplicativo representa uma resposta concreta às demandas contemporâneas da saúde comunitária, promovendo maior eficiência, segurança e impacto social na atenção primária.

2. Lista de requisitos funcionais

Os requisitos funcionais aqui descritos definem as funcionalidades que deverão ser implementadas no sistema, considerando o contexto acadêmico do desenvolvimento, podendo sofrer ajustes ou alterações ao longo do processo.

Tabela 1 - Lista de Requisitos Funcionais

Requisito	Descrição
REF01	O sistema deverá permitir o registro de novos usuários com perfis diferenciados (ACS, Supervisor, Administrador).
REF02	O sistema deverá autenticar usuários por meio de login e senha.
REF03	O sistema deverá permitir recuperação de senha via mecanismo seguro.
REF04	O sistema deverá permitir edição do perfil do usuário (nome, contato, microárea, função).
REF05	O sistema deverá suportar controle de permissões por função, restringindo o

	acesso a funcionalidades administrativas e de visualização.
REF06	O sistema deverá permitir desativação temporária de usuários (ex.: afastamento ou troca de microárea).
REF07	O sistema deverá disponibilizar formulários eletrônicos para registro de visitas domiciliares e atendimentos.
REF08	O aplicativo deverá manter uma fila de sincronização para envio e recebimento de dados quando a conectividade for restabelecida.
REF09	O aplicativo deverá exibir o status de sincronização por registro e permitir reenvio manual de itens com erro.
REF10	O sistema deverá aplicar mecanismos de resolução de conflitos, priorizando a última versão válida ou registrando conflito para revisão.
REF11	O aplicativo deverá permitir marcar e armazenar coordenadas geográficas de residências e pontos de interesse.
REF12	O aplicativo deverá exibir um mapa com os pontos da microárea do ACS.
REF13	O aplicativo deverá gerar rotas ou sequências de visita sugeridas para otimizar deslocamentos.
REF14	O aplicativo deverá suportar busca e filtragem de pontos por endereço, nome ou status da visita.
REF15	O aplicativo deverá permitir visualização offline de mapas.
REF16	O aplicativo deverá disponibilizar calendário de visitas e compromissos por ACS e microárea.
REF17	O sistema deverá permitir agendamento de visitas com definição de data e horário.
REF18	O aplicativo deverá enviar alertas e notificações locais (push/in-app) para visitas agendadas e pendências.
REF19	O sistema deverá permitir configuração de alertas personalizados por tipo de evento (vacinação, retorno, acompanhamento).
REF20	O aplicativo deverá permitir adiar, reagendar ou justificar visitas não realizadas.
REF21	O aplicativo deverá gerar relatórios sumarizados por microárea, ACS e período.
REF22	O aplicativo deverá apresentar dashboards com indicadores chave (visitas realizadas, pendências, cobertura).
REF23	O sistema deverá permitir exportação de relatórios em formato padrão (CSV ou similar), de acordo com o perfil de usuário, exibindo apenas dados de sua microárea ou subordinados.
REF24	O aplicativo deverá permitir filtragem por período, microárea e tipo de atendimento.
REF25	O aplicativo deverá permitir troca de mensagens entre ACS e população
REF26	O sistema deverá registrar histórico de comunicações associadas a usuários e microáreas.

REF27	O sistema deverá armazenar dados sensíveis de forma cifrada no dispositivo e em trânsito.
REF28	O sistema deverá permitir anonimização ou pseudonimização de dados para fins analíticos.
REF29	A interface deverá seguir guia de estilo com contraste adequado, legibilidade e navegação simples.
REF30	O aplicativo deverá oferecer feedback claro para ações críticas (sucesso, erro, instruções).
REF31	O sistema deverá incluir suporte para texto ampliado e alternativas visuais, quando aplicável.
REF32	O aplicativo deverá disponibilizar painel administrativo para gerenciamento de usuários, microáreas e parâmetros.
REF33	O sistema deverá permitir emissão de backups e restauração de dados locais e do servidor.
REF34	O sistema deverá permitir atualização remota do aplicativo e de artefatos configuráveis sem reinstalação manual.

Fonte: Os autores (2025).

3. Lista de requisitos não funcionais

Os requisitos não funcionais aqui descritos estabelecem as características de desempenho, usabilidade e qualidade do sistema, sujeitos a revisões e aprimoramentos conforme o desenvolvimento evoluir.

Tabela 2 - Lista de Requisitos Não Funcionais

Requisito	Descrição
RENF01	O aplicativo deverá apresentar resposta fluida nas principais telas e operações.
RENF02	As operações locais (salvar formulário, abrir lista) deverão ocorrer de forma fluida, dentro de poucos segundos.
RENF03	A sincronização deverá ser eficiente e não comprometer a experiência do usuário.
RENF04	O tempo de resposta deverá ser inferior a 4 segundos em ações simples e 1 minuto em sincronizações médias.
RENF05	O sistema deverá permanecer funcional em modo offline, permitindo anotações nas fichas e visualização de dados baixados anteriormente.
RENF06	O aplicativo deverá retomar automaticamente a sincronização ao restabelecer a conexão.
RENF07	A infraestrutura deverá manter boa disponibilidade e estabilidade durante os testes.

RENF8	O tratamento das informações deverá seguir princípios éticos e de confidencialidade.
RENF9	As interfaces deverão priorizar contraste, legibilidade e simplicidade.
RENF10	Os principais fluxos deverão ser concluídos em até três etapas.
RENF11	O aplicativo deverá ser compatível com recursos nativos de acessibilidade do Android.
RENF12	O aplicativo deverá ser compatível com as versões Android predominantes entre os dispositivos de teste.
RENF13	O backend deverá ser portável para execução local ou em nuvem sem grandes modificações.
RENF14	A fila de sincronização deverá persistir entre reinicializações.
RENF15	O sistema deverá identificar e sinalizar conflitos de dados para revisão.
RENF16	A arquitetura deverá ser modular e extensível para novos usuários e microáreas.
RENF17	O backend deverá suportar múltiplos usuários simultâneos sem degradação perceptível.
RENF18	Erros e exceções deverão ser registrados localmente ou enviados ao servidor.
RENF19	Indicadores de estabilidade e falhas deverão ser observados durante os testes.
RENF20	O sistema deverá permitir exportação e restauração de dados locais como forma de backup.
RENF21	O servidor deverá manter cópias de segurança periódicas com política de retenção definida.

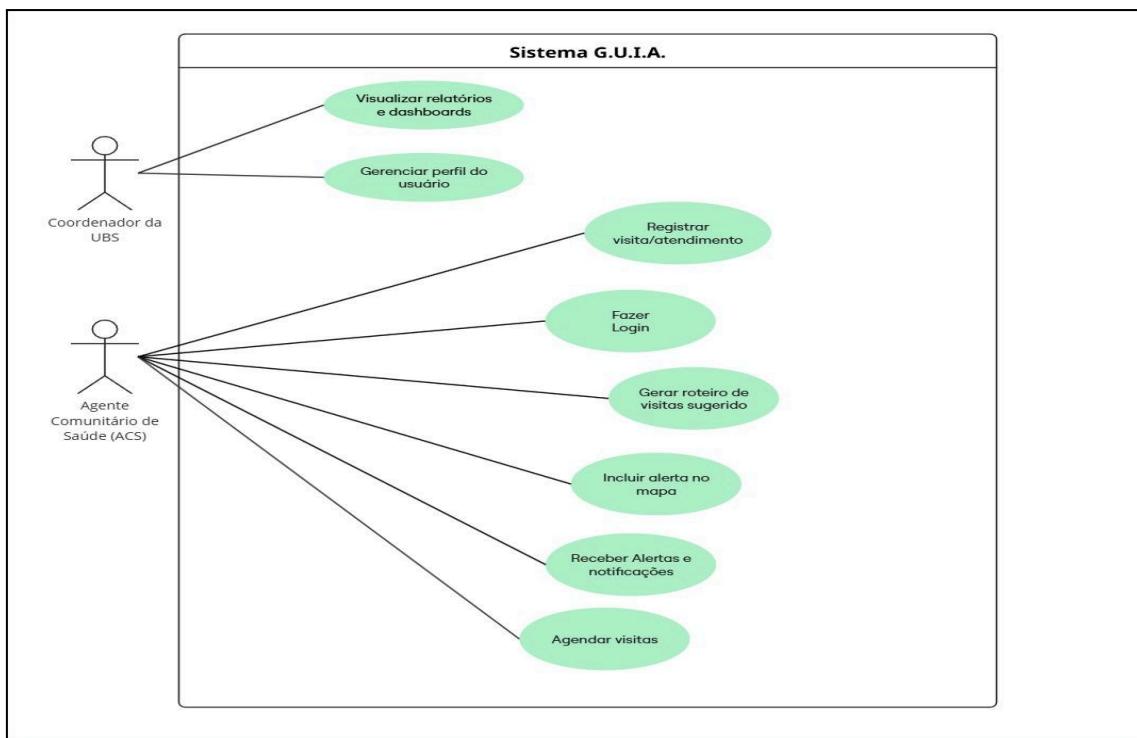
Fonte: Os autores (2025).

4. Especificação de casos de uso

Para apoiar a compreensão do funcionamento do Sistema G.U.I.A. e estruturar suas principais funcionalidades de forma clara, foram elaborados diagramas de caso de uso que representam, em diferentes níveis de detalhe, as interações entre os usuários e o sistema. Esses diagramas fornecem uma visão visual e organizada dos processos essenciais, permitindo identificar os atores envolvidos, seus objetivos e os fluxos de atividades que sustentam a operação da aplicação. A seguir, são apresentados os diagramas que compõem o modelo deste projeto, acompanhados de suas respectivas descrições.

O Diagrama 1 – Visão Geral do Sistema G.U.I.A. ilustra de forma simplificada o funcionamento do sistema e suas principais relações com os usuários e elementos externos. Ele fornece uma visão macro das funcionalidades e do contexto em que o sistema opera, servindo como ponto de partida para o entendimento geral do projeto.

Diagrama 1 - Visão Geral do Sistema G.U.I.A

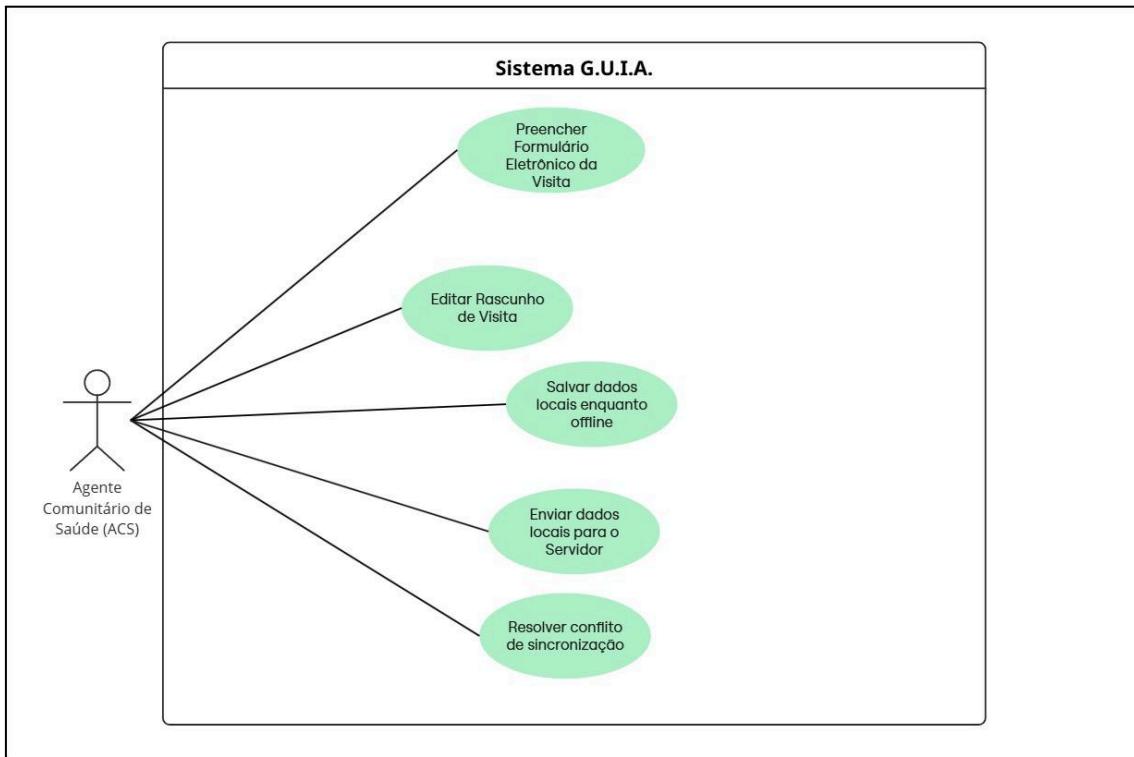


Fonte: Os autores (2025).

Esse diagrama oferece uma compreensão inicial do escopo da solução, permitindo identificar quem interage com o sistema, com quais objetivos e de que maneira essas interações estruturam o funcionamento geral da plataforma.

O Diagrama 2 – Coleta e Sincronização de Dados detalha o processo de registro de informações realizado pelo Agente Comunitário de Saúde (ACS) e sua posterior integração ao servidor remoto. Ele descreve o ciclo completo que se inicia na coleta em campo, segue pelo armazenamento local e culmina na sincronização automática quando houver conexão disponível.

Diagrama 2 - Coleta de e sincronização de dados

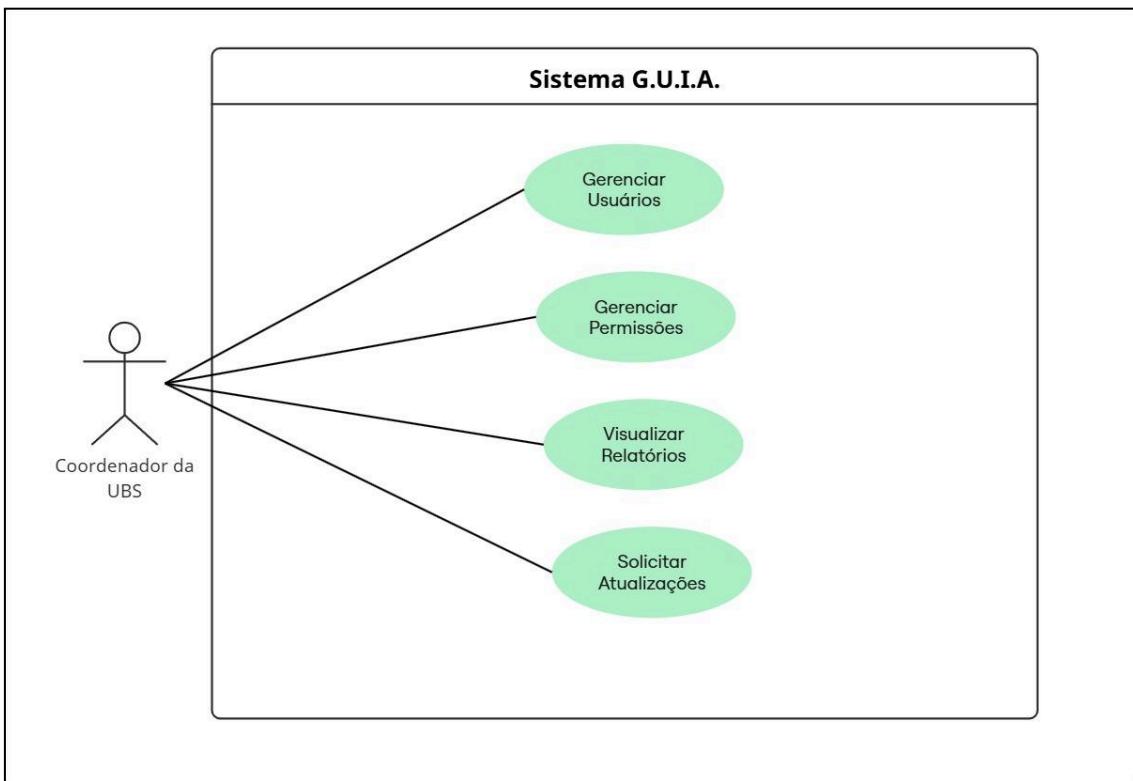


Fonte: Os autores (2025).

Na imagem é representado o fluxo de ações desde o momento da coleta de dados em campo até a sincronização com o servidor remoto, destacando como o sistema garante o funcionamento em modo offline e a posterior integração dos registros ao ambiente central. Esse diagrama auxilia na compreensão do ciclo completo de operação e comunicação entre o aplicativo local e o servidor.

O Diagrama 3 – Administração e Supervisão apresenta as principais funções de gestão, controle e acompanhamento realizadas pelos usuários administrativos do sistema G.U.I.A. Nesse diagrama são evidenciadas atividades como gerenciamento de acessos, manutenção e atualização de dados, além da supervisão das ações realizadas em campo. A representação demonstra como esses recursos oferecem suporte à tomada de decisão, fortalecem o controle operacional e contribuem para assegurar a integridade e a confiabilidade das informações registradas.

Diagrama 3 - Administração e Supervisão



Fonte: Os autores (2025).

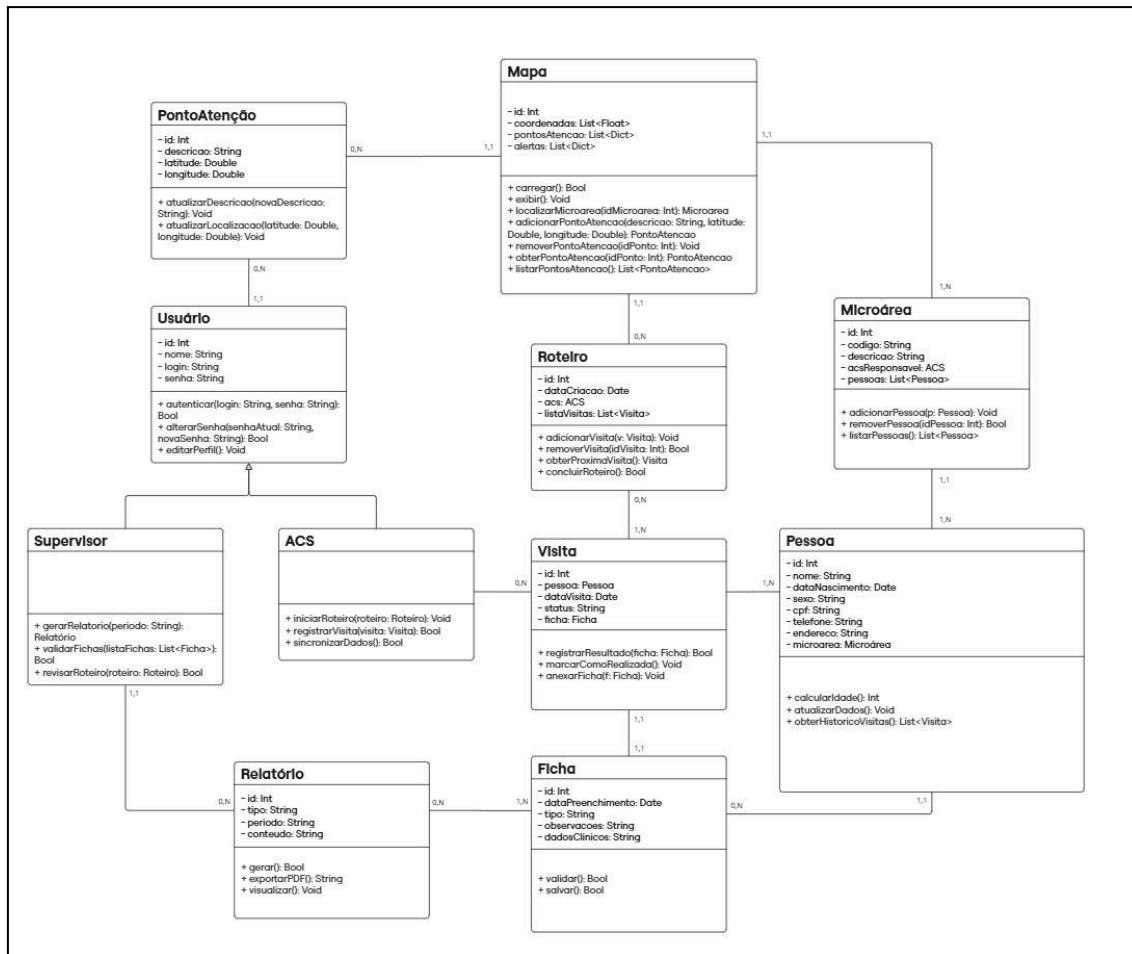
Em conjunto, os diagramas apresentados permitem visualizar de forma estruturada o funcionamento do Sistema G.U.I.A., desde as interações gerais até os processos específicos de coleta, sincronização e administração. Eles servem como base conceitual para as etapas posteriores do projeto, especialmente para o desenvolvimento do modelo de classes, da lógica de funcionamento e do planejamento dos testes. Assim, a seção cumpre o papel de consolidar a compreensão do comportamento esperado do sistema, garantindo que sua implementação siga uma referência clara, coerente e alinhada aos objetivos definidos.

5. Diagrama de Classes

O diagrama de classes representa a estrutura conceitual do sistema proposto, descrevendo as principais entidades, seus atributos, métodos e os relacionamentos entre elas. Ele foi elaborado com base na modelagem orientada a objetos e segue as convenções da UML (Unified Modeling Language). Através dessa representação, é possível visualizar a hierarquia de

classes, especializações, composições e dependências existentes, servindo como base para a implementação do sistema.

Diagrama 4 - Diagrama de Classes do Sistema G.U.I.A.



Fonte: Os autores (2025).

As classes “Usuário”, “Supervisor” e “ACS” representam os diferentes perfis do sistema, cada um com suas funções específicas. A classe “Mapa” centraliza informações geográficas e de localização, enquanto “PontoAtenção” e “Microárea” detalham a estrutura territorial de atuação. Já as classes “Pessoa”, “Visita”, “Roteiro”, “Ficha” e “Relatório” compõem o núcleo operacional, descrevendo as atividades de acompanhamento e registro de informações.

Esse modelo visa promover coerência, reuso de código e clareza na comunicação entre desenvolvedores e analistas, sendo uma etapa essencial para o desenvolvimento do sistema proposto.

6. Testes

A bateria de testes apresentada nesta seção estabelece uma cobertura funcional adequada às principais operações do aplicativo, contemplando autenticação, registro de visitas, operação offline, sincronização, persistência local e visualização de dados. Os casos de teste foram estruturados no formato de caixa-preta, focando exclusivamente no comportamento observável do sistema, sem levar em conta sua implementação interna, conforme recomendado em ambientes acadêmicos. O objetivo é assegurar que as funcionalidades atendam aos requisitos definidos nos diagramas de caso de uso e no modelo de classes, além de validar o comportamento geral do sistema do ponto de vista do usuário final.

Tabela 3 - Caso de Teste 01 – Login válido

Identificação	CT01	
Itens a testar	Autenticação com credenciais válidas	
Entradas	Campo	Valor
	Usuário	(usuário cadastrado)
Saídas Esperadas	Campo	Valor
	Tela	Tela inicial exibida
Ambiente	Banco de dados de teste	
Procedimentos	Inserir credenciais válidas e acessar o app	
Dependências	Usuário previamente cadastrado	

Fonte: Os autores (2025).

Tabela 4 - Caso de Teste 02 – Login inválido

Identificação	CT02	
Itens a testar	Autenticação com senha incorreta	
Entradas	Campo	Valor
	Usuário	(usuário cadastrado)
Saídas Esperadas	Campo	Valor
	Mensagem	Erro de autenticação
Ambiente	Banco de dados de teste	
Procedimentos	Inserir senha errada	
Dependências	Usuário previamente cadastrado	

Fonte: Os autores (2025).

Tabela 5 - Caso de Teste 03 – Registrar visita

Identificação	CT03	
Itens a testar	Registro de visita sem coordenadas	
Entradas	Campo	Valor
	Família	Selecionada
	Dados da visita	Preenchidos
Saídas Esperadas	Campo	Valor
	Registro	Armazenado localmente
Ambiente	Sessão ativa	
Procedimentos	Selecionar família, preencher dados e salvar	
Dependências	Família cadastrada	

Fonte: Os autores (2025).

Tabela 6 - Caso de Teste 04 – Registrar visita com coordenadas

Identificação	CT04	
Itens a testar	Registro de visita com coleta de localização	
Entradas	Campo	Valor
	Família	Selecionada
	Dados da visita	Preenchidos
Saídas Esperadas	Campo	Valor
	Localização	Latitude/Longitude capturadas
	Registro	Armazenado localmente
Dependências	Coordenadas	Armazenadas corretamente
	Família cadastrada	
	Sessão ativa	
Procedimentos	Selecionar família, preencher dados, capturar localização e salvar	

Fonte: Os autores (2025).

Tabela 7 - Caso de Teste 05 – Falha ao registrar visita

Identificação	CT05	
Itens a testar	Validação de campos obrigatórios	
Entradas	Campo	Valor
	Família	Selecionada
	Dados da visita	Incompletos
Saídas Esperadas	Campo	Valor
	Mensagem	Indicação dos campos faltantes
	Registro	Não armazenado
Ambiente	Sessão ativa	
Procedimentos	Tentar salvar sem preencher campos necessários	
Dependências	Família cadastrada	

Fonte: Os autores (2025).

Tabela 8 - Caso de Teste 06 – Salvar rascunho

Identificação	CT06	
Itens a testar	Armazenamento local como rascunho	
Entradas	Campo	Valor
	Família	Selecionada
Saídas Esperadas	Campo	Valor
	Status	"Rascunho"
Ambiente	Registro	
	Armazenado localmente	
Procedimentos	Sessão ativa	
Dependências	Interromper preenchimento e salvar	
Dependências		

Fonte: Os autores (2025).

Tabela 9 - Caso de Teste 07 – Sincronizar dados

Identificação	CT07	
Itens a testar	Sincronização com servidor	
Entradas	Campo	Valor
	Registros locais	Pendentes
Saídas Esperadas	Campo	Valor
	Status	Conexão
Ambiente	Ativa	
	Registrados enviados	
Procedimentos	Atualização	
Dependências	Status alterado para "sincronizado"	
Dependências		

Fonte: Os autores (2025).

Tabela 10 - Caso de Teste 08 – Falha de sincronização

Identificação	CT08	
Itens a testar	Comportamento em modo offline	
Entradas	Campo	Valor
	Conexão	Inativa
Saídas Esperadas	Campo	Valor
	Atualização	Falha de sincronização
Ambiente	Modo offline	
Procedimentos	Tentar sincronizar sem internet	
Dependências	Registros pendentes	

Fonte: Os autores (2025).

Tabela 11 - Caso de Teste 09 – Operação offline

Identificação	CT09	
Itens a testar	Registro de visita offline	
Entradas	Campo	Valor
	Conexão	Inativa
Saídas Esperadas	Campo	Valor
	Registro	Armazenado localmente
Ambiente	Status	
	“Pendente”	
Procedimentos	Modo offline	
Dependências	Registrar nova visita e salvar	

Fonte: Os autores (2025).

Tabela 12 - Caso de Teste 10 – Recuperar dados após reinício

Identificação	CT10	
Itens a testar	Persistência local	
Entradas	Campo	Valor
	Registros locais	Existentes
Saídas Esperadas	Campo	Valor
	Dados	Mantidos após reinício
Ambiente	Armazenamento local	
Procedimentos	Fechar e reabrir o app	
Dependências	Registros previamente salvos	

Fonte: Os autores (2025).

Tabela 13 - Caso de Teste 11 – Visualização de visitas

Identificação	CT11	
Itens a testar	Lista de visitas	
Entradas	Tipo de alerta	Valor
	Menu	Visitas previstas
Saídas Esperadas	Localização	Valor
	Lista	Exibição correta das visitas
Ambiente	Descrição	Valor
Procedimentos	Sessão ativa	
Dependências	Acessar menu de visitas	
Dependências	Registros existentes	

Fonte: Os autores (2025).

Tabela 14 - Caso de Teste 12 – Inserção de Alerta no Mapa

Identificação	CT12	
Itens a testar	Inserção de alerta no mapa	
Entradas	Campo	Valor
	Menu	Risco identificado
	Localização	Coordenadas capturadas
	Descrição	Texto informativo
Saídas Esperadas	Campo	Valor
	Mensagem	Alerta inserido com sucesso
	Mapa	Exibição do marcador no local informado
Ambiente	Sessão ativa + GPS habilitado	
Procedimentos	Acessar o mapa → Selecionar adicionar alerta → Preencher os campos → Confirmar operação	
Dependências	Permissão de localização + Conexão ativa ou armazenamento local disponível	

Fonte: Os autores (2025).

Dessa forma, o conjunto de testes aqui descrito fornece uma base suficiente para verificar o correto funcionamento do sistema e apoiar a identificação de erros antes de sua disponibilização final, promovendo maior qualidade e maturidade no produto desenvolvido. Caso testes adicionais sejam necessários em etapas posteriores, este plano poderá ser expandido de maneira incremental, preservando a estrutura já estabelecida.

7. Cronograma

A etapa seguinte do planejamento e execução de testes será realizada apenas no próximo período letivo, prevista para início em abril de 2026, quando será preparado o ambiente de testes, incluindo a configuração da base de dados, dispositivos e usuários de simulação.

A implementação formal dos casos de teste em planilhas ou ferramentas de controle ocorrerá na primeira quinzena de maio de 2026, seguida pela execução dos testes propriamente dita entre a segunda quinzena de maio e a primeira quinzena de junho de 2026. Essa fase contemplará a realização dos testes de caixa-preta, funcionais, interface, usabilidade interna, operação offline e sincronização.

As falhas encontradas serão corrigidas no decorrer da segunda quinzena de junho de 2026, incluindo a reexecução parcial dos testes afetados para validar os ajustes. Por fim, o relatório de avaliação final do ciclo de testes

será elaborado no final de junho de 2026, encerrando o processo com a consolidação dos resultados e o atestado de conformidade do sistema.

Tabela 15 - Cronograma de testes

Etapa	Atividade	Período	Descrição
Planejamento dos Testes	Definição dos tipos de teste, casos de teste, técnicas e critérios	Novembro de 2025	Finalização da documentação dos testes, especificações e critérios de aceitação.
Preparação do Ambiente	Configuração do ambiente de testes, base de dados de teste e dispositivos	Abril de 2026	Ajuste do ambiente simulado, configuração de usuários, microáreas e famílias.
Implementação dos Casos de Teste	Inserção dos cenários no sistema de acompanhamento	Maio de 2026	Registro e validação dos casos, checagem de coerência e vinculação aos requisitos.
Execução dos Testes	Execução manual dos testes de caixa-preta pelos alunos	Maio a Junho de 2026	Teste funcional, offline, sincronização, interface e aceitação. Registro de resultados e evidências.
Correções e Reexecução Parcial	Ajustes no aplicativo e reexecução dos testes impactados	Junho de 2026	Foco nas falhas encontradas; repetição seletiva dos casos afetados.
Avaliação Final e Relatório	Consolidação dos resultados e conclusão	Junho de 2026	Geração de relatório final com métricas, falhas, impactos e status do sistema.