Universidade dos Guararapes (UNIFG)

Ciências da Computação

UC: Modelos, métodos e técnicas da engenharia de software

Professor: Diógenes Carvalho Matias

Integrantes da equipe:

Brenno S. Montenegro Santos – 1352321600 – 7º Período

Fernando Antônio de A. N. Peixe – 1352317957 – 5º Período

Francisco Manoel da S. Neto – 13523110103 – 4º Período

Guilherme Andrade Ramos – 13523114778 – 5º Período

João Matheus F. Silva – 1352321353 – 4º Período

João Vitor Avelino da Silva – 1352322595 – 4º Período

RECIFE-PE

2025

Sumário

1. Introdução ao Projeto Granoo

O Granoo é um projeto de aplicativo móvel desenvolvido para simplificar e otimizar a experiência de compra de produtos de supermercado, bem como a logística de entrega. Desenvolvido com o framework Kivy, em linguagem Python, o Granoo visa oferecer uma plataforma intuitiva que conecta consumidores a supermercados parceiros, permitindo a visualização e seleção de produtos, a simulação de custos de entrega e o acompanhamento de pedidos.

Este documento apresenta uma visão geral da concepção do aplicativo, incluindo seus requisitos funcionais e não funcionais, arquitetura do sistema, modelos de dados e testes, além de diretrizes para segurança e conformidade com a LGPD. O foco está na entrega de uma solução viável, acessível e de fácil uso, especialmente pensada para usuários de dispositivos móveis.

2. Justificativa do Projeto

O aumento da demanda por conveniência e eficiência na rotina diária impulsionou significativamente o mercado de delivery e e-commerce. No contexto específico dos supermercados, a possibilidade de realizar compras de forma prática, com entrega flexível e controle total do processo, tornou-se um diferencial competitivo essencial. O projeto Granoo nasce com o propósito de atender a essa demanda por meio dos seguintes pontos:

- Conveniência e Otimização de Tempo: Ao permitir a realização de compras a qualquer momento e local, o aplicativo elimina deslocamentos e filas, devolvendo ao usuário o controle sobre seu tempo.
- Transparência e Economia: A funcionalidade de comparação de produtos e preços entre mercados, aliada à simulação de custos de entrega, promove escolhas mais conscientes e econômicas.
- Acompanhamento em Tempo Real: A possibilidade de seguir o status do pedido desde a seleção até a entrega agrega previsibilidade e confiança, características-chave para um bom serviço de delivery.
- Acessibilidade e Flexibilidade: Com entregas sob demanda ou agendadas, o Granoo se adapta a diferentes perfis de usuários e rotinas, ampliando o alcance da solução.
- Tecnologia Acessível e Multiplataforma: O uso do Kivy permite o desenvolvimento de uma única base de código para múltiplos sistemas operacionais (Android e iOS), reduzindo custos e acelerando o processo de entrega.
- Potencial de Expansão: Mesmo como uma versão simplificada, o Granoo tem espaço para evolução com integração de sistemas de pagamento reais, login seguro, ofertas personalizadas e novos parceiros comerciais.

Em resumo, o Granoo não é apenas um aplicativo de compras, mas uma proposta de solução tecnológica centrada na experiência do usuário, unindo simplicidade, praticidade e eficiência para o cotidiano do consumidor moderno.

3. Documento de Requisitos

3.1 Requisitos Funcionais (RF)

- RF001: Exibição de Boas-Vindas
    - O sistema deve exibir uma tela de boas-vindas ao iniciar.
    - A tela de boas-vindas deve conter opções para "Entrar" e "Cadastrar-se".
- RF002: Cadastro de Usuário
    - O sistema deve permitir que novos usuários se cadastrem, informando nome de usuário, senha e e-mail.
    - Após o cadastro, o sistema deve direcionar o usuário para a tela de login.
- RF003: Autenticação de Usuário (Login)
    - O sistema deve permitir que usuários existentes façam login com e-mail e senha.
    - Após o login bem-sucedido, o sistema deve direcionar o usuário para a tela inicial (Home).
    - O sistema deve apresentar a opção "Esqueceu a senha?".
- RF004: Visualização da Tela Inicial (Home)
    - O sistema deve exibir uma saudação ao usuário na tela inicial.
    - O sistema deve apresentar um campo de busca para produtos.
    - O sistema deve listar "Produtos em alta" em um formato de card horizontal e rolável.
    - Cada card de produto deve exibir nome, imagem e preço.
- RF005: Visualização de Detalhes do Produto
    - O sistema deve exibir uma tela de detalhes ao clicar em um card de produto na Home.
    - A tela de detalhes deve mostrar o nome, imagem e preço do produto selecionado.
    - A tela de detalhes deve ter um botão para retornar à Home.
- RF006: Navegação para Carteira

    - O sistema deve permitir que o usuário navegue para a tela de "Carteira" a partir da barra de navegação inferior.
- RF007: Visualização da Carteira
    - A tela de Carteira deve exibir o saldo atual do usuário (valor fixo para esta versão).
    - A tela de Carteira deve ter um botão para retornar à Home.
- RF008: Navegação para o Mapa
    - O sistema deve permitir que o usuário navegue para a tela de "Mapa" a partir da barra de navegação inferior.
- RF009: Visualização do Mapa
    - A tela do Mapa deve exibir um mapa interativo.
    - O mapa deve ser centralizado em uma localização padrão (São Paulo).
    - O mapa deve exibir um marcador no local centralizado.
- RF010: Busca de Local no Mapa
    - O sistema deve permitir que o usuário digite um endereço para buscar no mapa.
    - Ao buscar, o mapa deve ser centralizado em uma localização simulada próxima ao endereço digitado.
    - Um marcador deve ser adicionado na nova localização buscada, substituindo qualquer marcador anterior.
- RF011: Simulação de Preços de Entrega
    - Após a busca de um local no mapa, o sistema deve simular e exibir os preços de entrega para "Bike" e "Moto" com base em uma distância aleatória.
- RF012: Navegação Global
    - A barra de navegação inferior (Home, Carteira, Mapa) deve estar presente e funcional nas telas HomeScreen, CarteiraScreen e MapaScreen.

3.2 Requisitos Não Funcionais (RNF)

- RNF001: Usabilidade (UX)
    - A interface do usuário deve ser intuitiva e de fácil navegação para usuários de primeira viagem.
    - Os elementos interativos (botões, campos de texto) devem ser claramente distinguíveis e responsivos ao toque.

    - As transições entre as telas devem ser suaves.
- RNF002: Desempenho
    - O tempo de carregamento das telas deve ser inferior a 2 segundos em condições de rede normais.
    - A rolagem da lista de produtos deve ser fluida, sem travamentos.
    - A atualização do mapa e dos preços de entrega deve ser instantânea após a busca de um local.
- RNF003: Manutenibilidade
    - O código deve ser modular e bem comentado para facilitar futuras modificações e adições de funcionalidades.
    - A separação entre a lógica Python e o design KV deve ser mantida.
- RNF004: Portabilidade
    - O aplicativo deve ser executável em diferentes sistemas operacionais (Android, iOS) via Kivy, com adaptações mínimas.
- RNF005: Segurança (Básica)
    - Embora a autenticação seja simulada, em uma versão futura, as credenciais de login devem ser tratadas de forma segura (e.g., criptografia de senhas). (Para esta versão, é um placeholder).
- RNF006: Confiabilidade
    - O aplicativo deve ser estável e não apresentar falhas inesperadas durante o uso normal.
    - A funcionalidade do mapa e a adição/remoção de marcadores devem operar consistentemente.

---

4. Matriz de Rastreabilidade

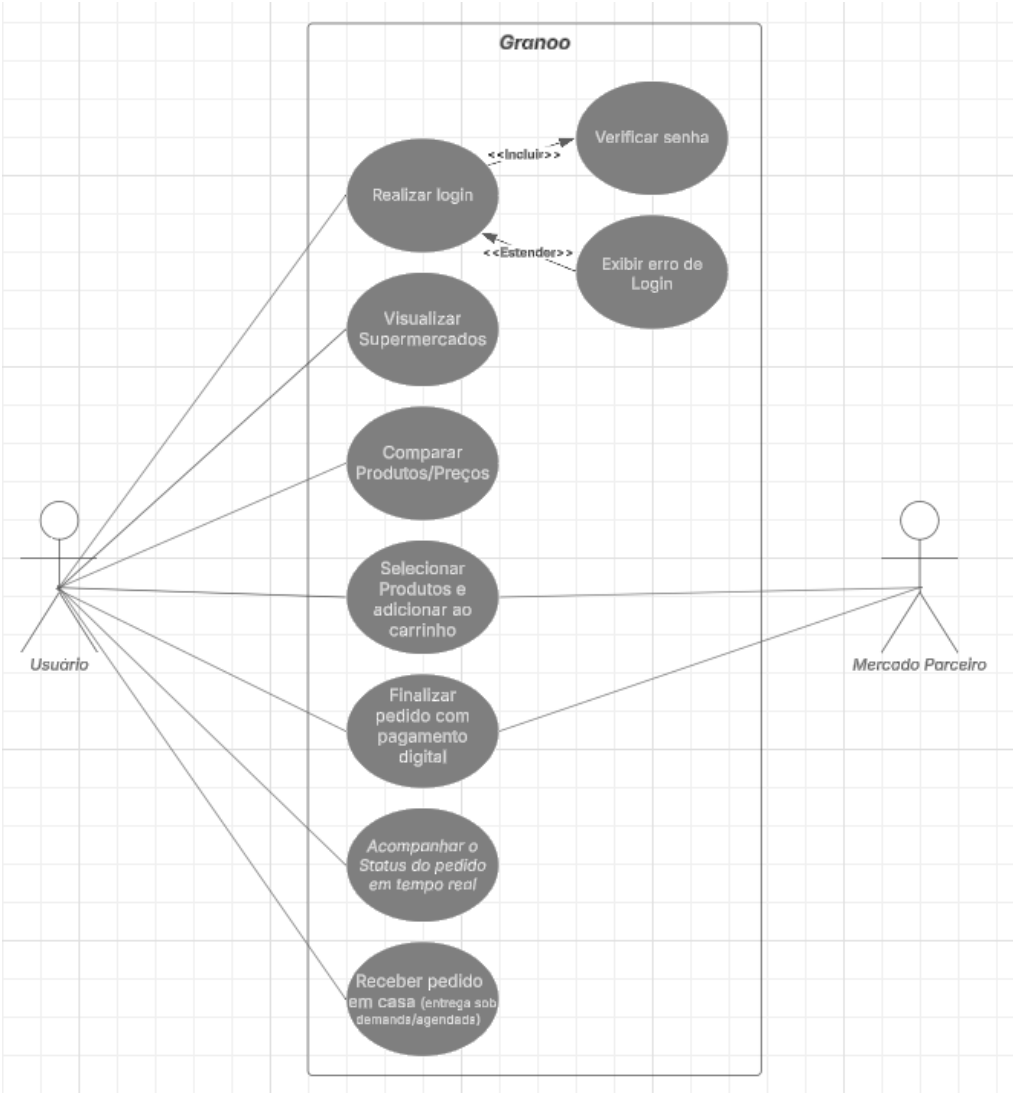
Esta matriz mapeia os requisitos funcionais aos casos de uso, entidades/módulos do código e exemplos de testes que os validariam.

Requisito	**Caso de Uso**	**Entidade(s) / Módulo(s)**	**Teste (Exemplo)**
RF001	CU001: Navegar para Boas-Vindas	BoasVindasScreen	Teste de UI: Exibição da tela de boas-vindas e botões.

RF002	CU002: Cadastrar Usuário	CadastroScreen	Teste de UI: Preenchimento e clique no botão de cadastro.
RF003	CU003: Autenticar Usuário	LoginScreen	Teste de UI: Preenchimento e clique no botão de login.
RF004	CU004: Visualizar Home	HomeScreen, ProdutoCard	Teste de UI: Exibição dos produtos, campo de busca.
RF005	CU005: Ver Detalhes do Produto	DetalhesProdutoScreen, ProdutoCard	Teste de Integração: Clique no card e verificação da tela de detalhes.
RF006	CU006: Acessar Carteira	HomeScreen (Bottom Bar)	Teste de UI: Clique no botão "Carteira".
RF007	CU007: Visualizar Carteira	CarteiraScreen	Teste de UI: Exibição do saldo e botão de voltar.
RF008	CU008: Acessar Mapa	HomeScreen (Bottom Bar)	Teste de UI: Clique no botão "Mapa".
RF009	CU009: Visualizar Mapa Inicial	MapaScreen, MapView, MapMarkerPopup	Teste de UI: Carregamento do mapa e marcador inicial.
RF010	CU010: Buscar Local no Mapa	MapaScreen (buscar_local)	Teste de Unidade: Função buscar_local com dados simulados.
RF011	CU011: Calcular Preços de Entrega	MapaScreen (buscar_local)	Teste de Unidade: Verificação dos valores gerados para preco_bike e preco_moto.

RF012	CU012: Navegar entre Telas Base	ScreenManager	Teste de UI: Transições entre Home, Carteira e Mapa.

5. Documento de Casos de Uso



Este diagrama ilustra como os usuários interagem com o sistema "Granoo" para alcançar seus objetivos. Ele é composto por Atores, Casos de Uso e Relacionamentos.

5.1. Atores

Atores são entidades (geralmente pessoas ou outros sistemas) que interagem com o sistema. Eles representam um papel que a entidade desempenha em relação ao sistema.

- Usuário:
    - Descrição: Representa o cliente final do aplicativo Granoo, ou seja, a pessoa que realiza compras, acompanha pedidos, e interage diretamente com a interface do usuário. É o ator principal e tem a maior parte das interações com o sistema.
    - Interações: Inicia a maioria dos casos de uso relacionados à compra e acompanhamento.

- Mercado Parceiro:
    - Descrição: Representa um supermercado ou estabelecimento comercial que é parceiro do Granoo. Este ator provavelmente interage com o sistema para gerenciar produtos, preços e receber/processar pedidos. Embora o diagrama não mostre explicitamente todos os casos de uso para o Mercado Parceiro, sua presença indica uma interface ou conjunto de funcionalidades dedicadas a ele.
    - Interações: Interage diretamente com o caso de uso "Selecionar Produtos e adicionar ao carrinho", o que sugere que o Mercado Parceiro é o fornecedor dos produtos. Em um sistema real, ele também estaria envolvido na atualização de inventário, confirmação de pedidos, etc.

5.2. Casos de Uso

Casos de Uso são as funcionalidades que o sistema oferece para atingir um objetivo específico do ator. Cada caso de uso representa uma ação ou conjunto de ações que o sistema realiza em resposta à interação de um ator.

- Realizar login:
    - Descrição: Permite que um usuário existente acesse sua conta no aplicativo, fornecendo suas credenciais (usuário e senha). É a porta de entrada para a maioria das funcionalidades do aplicativo.
    - Ator Primário: Usuário.
    - Relacionamentos:
        - <<Incluir>> Verificar senha: Indica que o caso de uso "Verificar senha" *sempre* faz parte do "Realizar login". O sistema precisa validar as credenciais fornecidas pelo usuário.
        - <<Estender>> Exibir erro de Login: Indica que o caso de uso "Exibir erro de Login" *pode acontecer* durante o "Realizar login" sob uma condição específica (por exemplo,

credenciais incorretas). Não é uma parte obrigatória do fluxo de sucesso.

- Verificar senha:
    - Descrição: Sub-processo interno do sistema que valida se a senha fornecida pelo usuário corresponde à senha registrada para aquele usuário.
    - Ator Primário: Nenhum (é um processo interno disparado por "Realizar login").
    - Relacionamento: Inclusão (<<Incluir>>) do "Realizar login".

- Exibir erro de Login:
    - Descrição: O sistema informa ao usuário que a tentativa de login falhou, geralmente devido a credenciais inválidas.
    - Ator Primário: Nenhum (é um processo interno disparado por "Realizar login").
    - Relacionamento: Extensão (<<Estender>>) de "Realizar login".

- Visualizar Supermercados:
    - Descrição: O usuário pode navegar e ver uma lista ou mapa de supermercados parceiros disponíveis no Granoo.
    - Ator Primário: Usuário.

- Comparar Produtos/Preços:
    - Descrição: Permite ao usuário visualizar e comparar produtos e seus respectivos preços entre diferentes supermercados parceiros.
    - Ator Primário: Usuário.

- Selecionar Produtos e adicionar ao carrinho:
    - Descrição: O usuário escolhe os produtos desejados nas listas de supermercados e os adiciona a um carrinho de compras virtual. Este é um caso de uso central para a funcionalidade de compra.

    - Atores: Usuário (inicia a seleção) e Mercado Parceiro (fornece os produtos para seleção).

- Finalizar pedido com pagamento digital:
    - Descrição: O usuário conclui a compra de itens no carrinho, efetuando o pagamento através de métodos digitais integrados no aplicativo.
    - Ator Primário: Usuário.

- Acompanhar o Status do pedido em tempo real:
    - Descrição: O usuário pode verificar o progresso do seu pedido após a finalização, desde a preparação no supermercado até a entrega.
    - Ator Primário: Usuário.

- Receber pedido em casa (entrega sob demanda/agendada):
    - Descrição: O usuário recebe fisicamente o pedido em seu endereço, podendo ter optado por entrega imediata ("sob demanda") ou agendada para um horário específico.
    - Ator Primário: Usuário.

5.3. Relacionamentos

Os relacionamentos indicam como os casos de uso se interconectam ou como os atores se associam aos casos de uso.

- Associação: Linha contínua entre um ator e um caso de uso. Indica que o ator interage com aquele caso de uso para atingir um objetivo.
- <<Incluir>> (Include): Representado por uma seta tracejada com a estereótipo <<include>>. Significa que um caso de uso base *sempre* incorpora a funcionalidade de outro caso de uso (o caso de uso "incluído"). O caso de uso incluído é essencial para o sucesso do caso de uso base.
    - Exemplo no Diagrama: Realizar login <<Incluir>> Verificar senha.
        - Para realizar um login, a verificação da senha *precisa* acontecer.

- <<Estender>> (Extend): Representado por uma seta tracejada com a estereótipo <<extend>>. Significa que um caso de uso base *pode* ter sua funcionalidade estendida por outro caso de uso (o caso de uso "extensor") sob certas condições. O caso de uso extensor é opcional e ocorre apenas quando a condição é atendida.
    - Exemplo no Diagrama: Realizar login <<Estender>> Exibir erro de Login.
        - O login pode falhar, e *se falhar*, o erro de login será exibido. A exibição do erro não é uma parte obrigatória de *todo* processo de login (apenas dos que falham).

6. Modelagem do Sistema

6.1. Diagrama Entidade-Relacionamento (ER)

No contexto deste aplicativo Kivy, que opera principalmente na interface do usuário e não possui um banco de dados persistente ou comunicação com um backend complexo, as "entidades" são mais conceituais e representam os objetos de dados que o aplicativo manipula internamente.

- Produto:
    - Atributos: nome (String), imagem (URL String), preco (String)
- Usuário:
    - Atributos: nome_usuario (String), senha (String), email (String)
    - *Observação:* Estes dados são apenas coletados na UI, não são armazenados ou validados em um banco de dados real.

Relacionamentos:

- Não há relacionamentos complexos entre essas entidades dentro do escopo atual do aplicativo, pois não há persistência ou interconexão de dados entre elas. O "Produto" é uma entidade de exibição, e "Usuário" é uma entidade de entrada de dados.

6.2. Diagrama Lógico

O diagrama lógico representa a estrutura de classes e como elas interagem no nível do código Python.

6.3. Diagrama Físico

O diagrama físico descreve como o aplicativo é executado no ambiente de hardware e software, focando nos componentes principais e suas interações.

7. Documentação de Testes

7.1 Plano de Testes

- Objetivo: Validar a conformidade do aplicativo "Granoo" com os requisitos funcionais e não funcionais estabelecidos, garantindo sua usabilidade, desempenho e estabilidade.

- Estratégia de Testes:
  - Testes de Unidade: Foco na validação de componentes de código isolados e funções específicas (ex: cálculo de preços na MapaScreen, atualização de propriedades no ProdutoCard).
  - Testes de Integração: Verificação das interações entre diferentes módulos e telas (ex: transição de login para home, exibição de detalhes do produto ao clicar em um card).
  - Testes de Sistema: Avaliação do aplicativo como um todo, simulando cenários de uso real para garantir que todas as funcionalidades operem em conjunto conforme o esperado (ex: fluxo completo de cadastro, login, visualização de produtos e busca no mapa).
  - Testes de Usabilidade (UX): Avaliação da experiência do usuário, incluindo facilidade de uso, clareza da interface e feedback visual.
  - Testes de Desempenho: Verificação da responsividade da interface, tempo de carregamento de telas e fluidez das animações/rolagem.

- Ferramentas Sugeridas:
  - Testes de Unidade/Integração: unittest (módulo padrão do Python) para lógica Python pura, embora testes de UI Kivy sejam mais desafiadores de automatizar sem ferramentas dedicadas.
  - Testes de UI/Sistema: Principalmente testes manuais e de observação devido à natureza do Kivy para interfaces de usuário, mas ferramentas de captura de tela ou gravação de interação podem auxiliar na documentação de bugs.

- Ambiente de Teste:

- Software: Python 3.x, Kivy Framework, Kivy Garden (MapView).
- Hardware: Emuladores de dispositivos móveis (Android Studio, Xcode Simulator) ou dispositivos físicos (smartphones, tablets).

7.2 Casos de Testes

Código Caso de Teste	**Descrição do Caso de Teste**	**Entrada/Passos**	**Resultado Esperado**	**Requisito(s)**
TC001	Acessar Tela de Login pela Boas-Vindas	1. Abrir o aplicativo Granoo. 2. Clicar no botão "Entrar" na tela de Boas-Vindas.	A tela de Login é exibida.	RF001
TC002	Acessar Tela de Cadastro pela Boas-Vindas	1. Abrir o aplicativo Granoo. 2. Clicar no botão "Não tem uma conta? Cadastre-se" na tela de Boas-Vindas.	A tela de Cadastro é exibida.	RF001
TC003	Realizar Cadastro de Novo Usuário	1. Na tela de Cadastro, preencher "Nome de Usuário", "Senha" e "E-mail".	O aplicativo redireciona para a tela de Login.	RF002

		2. Clicar em "Cadastrar".		
TC004	Retornar da Tela de Cadastro	1. Na tela de Cadastro, clicar no botão "Voltar".	O aplicativo redireciona para a tela de Boas-Vindas.	RF002
TC005	Realizar Login Válido (Simulado)	1. Na tela de Login, preencher "E-mail" e "Senha". 2. Clicar em "Login".	O aplicativo redireciona para a tela Home com a saudação "Olá, Usuário!".	RF003, RF004
TC006	Acessar "Esqueceu a senha?"	1. Na tela de Login, clicar no botão "Esqueceu a senha?".	O botão responde (mesmo que não haja funcionalidade de recuperação de senha).	RF003
TC007	Visualizar Produtos em Destaque na Home	1. Após o login, observar a tela Home.	A lista de "Produtos em alta" é exibida horizontalmente com pelo menos 5 cards, cada um com imagem, nome e preço.	RF004
TC008	Usar Campo de Busca (Home)	1. Na tela Home, digitar "Maçã" no campo	A função busca_changed é acionada (para futura filtragem). (Nesta versão,	RF004

		"Procurar produtos...". 2. Pressionar Enter/Sair do campo.	não há efeito visual de filtro).	
TC009	Visualizar Detalhes de um Produto	1. Na tela Home, clicar em um card de produto (ex: "Maçã").	A tela de Detalhes do Produto é exibida, mostrando a imagem, nome e preço da "Maçã".	RF005
TC010	Retornar da Tela de Detalhes do Produto	1. Na tela de Detalhes do Produto, clicar no botão "Voltar".	O aplicativo redireciona para a tela Home.	RF005
TC011	Navegar para a Carteira	1. Na tela Home, clicar no botão "Carteira" na barra inferior.	A tela "Carteira" é exibida, mostrando o saldo "R$ 123,45".	RF006, RF007
TC012	Retornar da Tela da Carteira	1. Na tela "Carteira", clicar no botão "Voltar".	O aplicativo redireciona para a tela Home.	RF007
TC013	Navegar para o Mapa	1. Na tela Home, clicar no botão "Mapa" na barra inferior.	A tela "Mapa" é exibida, mostrando o mapa centralizado em São Paulo	RF008, RF009

			com um marcador.	
TC014	Buscar Local no Mapa	1. Na tela "Mapa", digitar "Avenida Paulista" no campo "Digite o local". 2. Clicar em "Buscar".	O mapa é recentralizado em uma nova localização simulada, e o marcador é atualizado. Os preços "Preço Bike" e "Preço Moto" são exibidos com valores simulados.	RF010, RF011
TC015	Retornar da Tela do Mapa	1. Na tela "Mapa", clicar no botão "Voltar".	O aplicativo redireciona para a tela Home.	RF009
TC016	Verificação da Barra de Navegação Inferior	1. Navegar entre Home, Carteira e Mapa usando os botões da barra inferior.	A barra de navegação inferior permanece visível e funcional em todas as telas esperadas (Home, Carteira, Mapa).	RF012

7.3 Critérios de Aceitação

Os critérios de aceitação são as condições que uma funcionalidade deve atender para ser considerada concluída e correta. Eles são derivados diretamente dos requisitos funcionais e não funcionais.

- RF001 (Boas-Vindas):

    - A tela de boas-vindas é a primeira tela exibida ao iniciar o aplicativo.
    - Os botões "Entrar" e "Cadastrar-se" são visíveis, clicáveis e redirecionam para as telas corretas.
- RF002 (Cadastro):
    - O usuário pode preencher todos os campos (Nome de Usuário, Senha, E-mail).
    - Ao clicar em "Cadastrar", o aplicativo navega para a tela de Login.
    - O botão "Voltar" redireciona para a tela de Boas-Vindas.
- RF003 (Login):
    - O usuário pode preencher os campos de E-mail e Senha.
    - Ao clicar em "Login", o aplicativo navega para a tela Home.
    - O botão "Esqueceu a senha?" é visível (mesmo que não funcional nesta versão).
- RF004 (Home):
    - A saudação "Olá, Usuário!" é exibida no cabeçalho.
    - O campo "Procurar produtos..." está visível e permite entrada de texto.
    - A lista de "Produtos em alta" exibe pelo menos 5 cards de produtos.
    - Os cards de produtos são roláveis horizontalmente.
    - Cada card exibe corretamente a imagem, nome e preço do produto.
- RF005 (Detalhes do Produto):
    - Ao clicar em um ProdutoCard, a tela de Detalhes do Produto é exibida.
    - A tela de detalhes exibe a imagem, nome e preço correspondentes ao produto clicado.
    - O botão "Voltar" na tela de detalhes retorna para a Home.
- RF006, RF007 (Carteira):
    - Ao clicar em "Carteira" na barra inferior, a tela de Carteira é exibida.
    - A tela de Carteira exibe o texto "Saldo atual: R$ 123,45".
    - O botão "Voltar" na tela de Carteira retorna para a Home.
- RF008, RF009 (Mapa):
    - Ao clicar em "Mapa" na barra inferior, a tela de Mapa é exibida.

    - O mapa é carregado e centralizado na latitude/longitude padrão (-23.55052, -46.633308).
    - Um marcador é visível no centro do mapa.
- RF010, RF011 (Busca e Preços no Mapa):
    - O campo "Digite o local" permite entrada de texto.
    - Ao clicar em "Buscar", o mapa é centralizado em uma nova localização (simulada).
    - Um novo marcador é adicionado na nova localização, e o marcador anterior é removido.
    - Os campos "Preço Bike" e "Preço Moto" são atualizados com valores numéricos formatados como "R$ X,XX".
- RNF001 (Usabilidade):
    - Todas as transições de tela são suaves e sem atrasos perceptíveis.
    - Todos os botões e campos de entrada respondem imediatamente ao toque/clique.
    - A rolagem da lista de produtos na Home é fluida.
- RNF002 (Desempenho):
    - O aplicativo é carregado em menos de 5 segundos na inicialização.
    - As telas são carregadas em menos de 2 segundos após a transição.

8. Documentação de Segurança

8.1 Autenticação e Autorização

- Status Atual: O aplicativo "Granoo" implementa um sistema de autenticação e cadastro *simulado*. Isso significa que os dados de login e cadastro são coletados, mas não são validados contra um banco de dados real, nem há persistência de sessão ou controle de acesso baseado em papéis. Qualquer combinação de e-mail/senha permite o "login" e a navegação para a tela principal.
- Melhorias Futuras Sugeridas:
    - Backend de Autenticação: Integrar com um serviço de autenticação robusto (ex: Firebase Authentication, AWS Cognito, Auth0) ou um backend customizado com APIs seguras.
    - Validação de Credenciais: Implementar validação rigorosa das credenciais do usuário no servidor (hash de senhas com sal, comparação segura).

- Gerenciamento de Sessão: Utilizar tokens (ex: JWT) para gerenciar sessões de usuário, garantindo que o acesso a funcionalidades restritas seja autorizado após o login.
- Controle de Acesso (Autorização): Se houver funcionalidades diferenciadas para diferentes tipos de usuários (ex: admin, cliente, entregador), implementar um sistema de autorização para restringir o acesso a recursos específicos.

8.2 Criptografia

- Dados em Trânsito:
  - Atualmente, as imagens dos produtos e os tiles do mapa são carregados via URLs HTTPS, o que garante a criptografia da comunicação entre o aplicativo e os servidores externos (usando SSL/TLS).
  - Para futuras integrações com APIs de backend (ex: para login, cadastro, dados de usuário), o uso exclusivo de HTTPS/SSL/TLS é mandatório para proteger todos os dados em trânsito contra interceptação.

- Dados em Repouso:
  - O aplicativo "Granoo", em sua versão atual, não armazena dados sensíveis localmente de forma persistente. Os dados de usuário inseridos no cadastro/login e os produtos carregados são voláteis e existem apenas enquanto o aplicativo está em execução.
  - Melhorias Futuras Sugeridas: Se houver necessidade de persistir dados sensíveis localmente (ex: informações de perfil, histórico de pedidos, tokens de sessão), deve-se considerar a criptografia desses dados utilizando APIs de armazenamento seguro do sistema operacional (ex: Android KeyStore, iOS Keychain) ou bibliotecas de criptografia para bancos de dados locais (ex: SQLCipher para SQLite).

8.3 Armazenamento Seguro

- Status Atual: Conforme mencionado, o aplicativo não persiste dados sensíveis do usuário. Os dados de produtos são carregados estaticamente ou via URLs diretas. Não há um banco de dados local ou sistema de arquivos que exija segurança de armazenamento.

- Melhorias Futuras Sugeridas:

- Credenciais e Tokens: Nunca armazenar senhas em texto puro. Armazenar tokens de sessão e credenciais de forma segura usando as APIs de armazenamento de credenciais fornecidas pelo sistema operacional (KeyStore/Keychain).
- Dados do Usuário: Se dados como histórico de pedidos, endereços salvos ou informações de pagamento forem armazenados localmente, eles devem ser criptografados e armazenados em diretórios de aplicativo seguros que não sejam facilmente acessíveis por outras aplicações.
- Dados de Configuração: Configurações não sensíveis podem ser armazenadas em arquivos de preferência.

8.4 Proteção contra Ameaças

- Validação de Entrada:
  - Embora o Kivy forneça a interface para entrada de dados, o código atual não implementa validações robustas (ex: para formatos de e-mail, força de senha, sanitização de texto livre).
  - Melhorias Futuras Sugeridas: Implementar validação de entrada rigorosa no lado do cliente (para usabilidade) e, crucialmente, no lado do servidor (para segurança) para prevenir ataques como Injeção de SQL (se um banco de dados for usado), Cross-Site Scripting (XSS, se houver conteúdo web view ou gerado pelo usuário), e outros.

- Vulnerabilidades Comuns em Aplicativos Móveis (Considerações Futuras):
  - Insecure Data Storage: Evitar armazenar dados sensíveis em locais não criptografados ou facilmente acessíveis.
  - Insecure Communication: Sempre usar HTTPS/TLS para todas as comunicações com servidores.
  - Improper Authentication/Authorization: Garantir que o processo de login e controle de acesso seja robusto e testado.
  - Code Tampering/Reverse Engineering: Para aplicativos de produção, ofuscação de código e técnicas anti-tampering podem ser consideradas para dificultar a engenharia reversa do APK/IPA.
  - Insecure Third-Party Dependencies: Manter as bibliotecas de terceiros (como Kivy Garden MapView) atualizadas para evitar vulnerabilidades conhecidas.

8.5 Conformidade com LGPD (Lei Geral de Proteção de Dados)

- Coleta de Dados:
    - Atualmente, o aplicativo coleta o nome de usuário, e-mail e senha durante o processo de cadastro (simulado).
    - Para conformidade com a LGPD, o aplicativo deve:
        - Informar claramente ao usuário quais dados estão sendo coletados.
        - Obter consentimento explícito e específico para a coleta e tratamento de dados pessoais.
        - Informar a finalidade da coleta de cada dado.

- Consentimento:
    - Deve haver uma tela de "Termos de Uso e Política de Privacidade" de fácil acesso, onde o usuário possa consentir com o tratamento de seus dados.
    - O consentimento deve ser livre, informado e inequívoco.

- Direitos do Titular:
    - O aplicativo (ou seu backend associado) deve fornecer mecanismos para que os usuários possam exercer seus direitos como titulares dos dados, incluindo:
        - Acesso: Solicitar uma cópia dos seus dados pessoais.
        - Retificação: Corrigir dados incorretos ou desatualizados.
        - Exclusão: Solicitar a exclusão dos seus dados (Direito ao Esquecimento).
        - Portabilidade: Receber seus dados em um formato estruturado para transferir para outro serviço.
        - Revogação do Consentimento: Retirar o consentimento a qualquer momento.

- Segurança dos Dados:

    - Implementar medidas técnicas e organizacionais adequadas para proteger os dados pessoais contra acessos não autorizados, vazamentos, alterações ou destruição. Isso inclui criptografia, controle de acesso, auditorias e treinamento de equipe.

- Retenção de Dados:
    - Definir e comunicar uma política clara sobre por quanto tempo os dados pessoais serão armazenados e quando serão descartados de forma segura