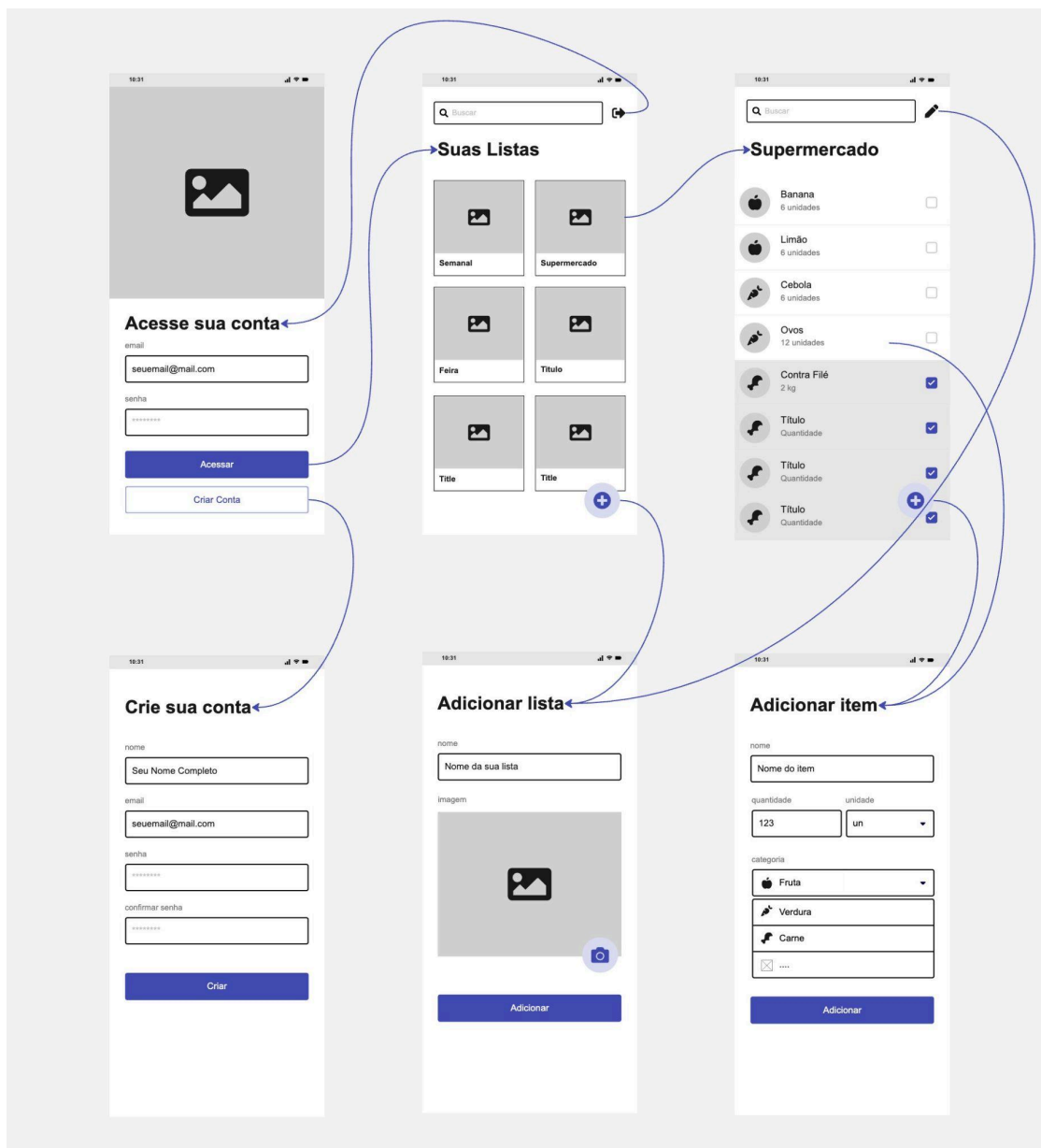


Projeto Prático (Parcial)

Objetivo

Desenvolver um **Aplicativo de Lista de Compras** para Android Nativo com Kotlin.

Protótipos



Contextualização

O Aplicativo de Lista de Compras é projetado para ajudar os usuários a criar e gerenciar listas de compras de forma eficiente. Ele permite que os usuários adicionem itens à lista, marquem itens como comprados e removam itens conforme necessário.

Requisitos Funcionais

RF001	Login / Logout
	<p>O aplicativo deve ter uma tela de login com campos de email, senha, e o logotipo da aplicação.</p> <p>Validar se o email é válido e se os campos não estão vazios. Credenciais simuladas (dados mocados) podem ser usadas.</p> <p>Deve haver botões para login e cadastro de novo usuário.</p> <p>Ao realizar logout, os dados devem ser descartados e a tela de login reapresentada.</p>
RF002	Cadastro de Usuário
	<p>O aplicativo deve permitir o cadastro de novos usuários com os campos: nome, e-mail, senha e confirmação de senha.</p> <p>Validar preenchimento de todos os campos, se o e-mail é válido, e se senha e confirmação coincidem.</p>
RF003	Gestão de Listas de Compras
	<p>O aplicativo deve permitir criar, editar, listar e excluir listas de compras com título e imagem (opcional), possibilitando selecionar uma mídia de imagem do dispositivo.</p> <p>A listagem deve ser ordenada alfabeticamente e exibir título e imagem. Caso não haja imagem, exibir um placeholder.</p>

	Ao excluir uma lista, todos os itens associados também devem ser excluídos.
--	---

RF004	Gestão de Itens da Lista
	<p>O aplicativo deve permitir criar, editar, listar e excluir itens em listas de compras.</p> <p>Cada item deve incluir nome, quantidade, unidade e categoria. Por exemplo, 2 un de maçãs (fruta), 2 kg de frango (carne), etc.</p> <p>Os itens devem ser apresentados dentro de suas respectivas listas de compras, exibindo nome, quantidade, unidade e o ícone da categoria. Devem estar em ordem alfabética e agrupados por categoria.</p> <p>Deve ser possível marcar itens que já foram comprados, assim como desmarcar. Os itens marcados devem ficar separados (abaixo) dos demais.</p>

RF005	Buscas
	Permitir pesquisa por listas de compras e itens dentro das listas. A pesquisa poderá ser realizada a partir do nome do item ou da lista.

Requisitos Não-Funcionais

- O aplicativo deve ser desenvolvido com o Android Nativo com Kotlin.
- O aplicativo poderá ser desenvolvido em grupos de até 3 pessoas.
- O design deve ser intuitivo e fácil de usar, com botões e controles claramente identificáveis, seguindo as especificações do [Material Design](#).
- A navegação deve ser fluida e seguir as especificações do [Material Design](#).
- Deve haver tratamento para que não haja perda de dados com mudanças de configurações do dispositivo (por exemplo, rotação de tela).
- Deve-se utilizar o recurso de vinculação de visualizações ([ViewBinding](#)).

Importante

O aplicativo deverá ser capaz de manipular dados de forma dinâmica. Assim sendo, na funcionalidade de *Criar Lista*, por exemplo, os dados precisarão ser armazenados em memória.

O que deverá ser entregue?

- A entrega será realizada através de um repositório público no GitHub. Para isso, você deverá criar um repositório e hospedar o código fonte completo do aplicativo. Entregar um arquivo texto com nome e código dos alunos envolvidos, assim como link do projeto no Github.
- Deve ser entregue também um vídeo de até 5 minutos apresentando o aplicativo. A apresentação deve abordar o aplicativo em funcionamento e também seu código. Na apresentação do código, focar nas partes relacionadas às rubricas. O vídeo deve estar no formato MP4.

Critérios de Avaliação

[10%] RF001 Login / Logout	Pontuação
<input type="checkbox"/> A tela de login contém campos de e-mail, senha, logotipo e botões de "Entrar" e "Cadastrar". <input type="checkbox"/> Validação correta de e-mail (formato) e campos vazios. <input type="checkbox"/> O botão "Entrar" aciona o login, e "Cadastrar" navega para o cadastro de usuário. <input type="checkbox"/> O logout redireciona para a tela de login, descartando dados.	
[10%] RF002 Cadastro de Usuário	Pontuação
<input type="checkbox"/> Cadastro de usuário possui todos os campos necessários <input type="checkbox"/> Validação correta de nome, e-mail, senha e confirmação de senha.	
[30%] RF003 Gestão de Listas de Compras	Pontuação
<input type="checkbox"/> Criação, edição e exclusão de listas funcionando, com dados mantidos em memória corretamente. <input type="checkbox"/> Validação do nome da lista não vazio.	

<input type="checkbox"/> Permissões para capturar ou escolher e exibir imagem. <input type="checkbox"/> Listagem exibida com uma RecyclerView. <input type="checkbox"/> Exclusão de lista remove também dados de seus itens.	
--	--

[40%] RF004 Gestão de Itens da Lista	Pontuação
<input type="checkbox"/> Adição, edição e remoção de itens funcionando, com dados mantidos em memória corretamente. <input type="checkbox"/> Validação de informações dos itens para o cadastro. <input type="checkbox"/> Listagem exibida com uma RecyclerView. <input type="checkbox"/> Itens podem ser marcados como comprados e indicados visualmente na interface.	

[10%] RF005 Buscas	Pontuação
<input type="checkbox"/> Pesquisa de listas e itens funcionando, com filtragem eficiente.	

Considerações Finais

- A identificação de cópias indiscriminadas de código fonte da Internet, IA ou de colegas da turma, ocasionará a perda da pontuação. Caso haja suspeita de fraude, o grupo poderá passar por uma avaliação oral acerca do projeto desenvolvido.
- A entrega fora do prazo estipulado acarretará em nota zero. O repositório no Github será avaliado levando em consideração somente os commits realizados até a data de entrega. **Sugestão:** tente baixar seu código em outra máquina e executar a aplicação para ter a certeza de que não está faltando nenhum arquivo de configuração que possa vir a prejudicar sua nota.