

# AppMercado

## [Descrição](#)

## [Público-Alvo/Intended User](#)

## [Funcionalidades/Features](#)

## [Protótipo de Interfaces do Usuário](#)

### [Telas](#)

## [Considerações Chave/Key Considerations](#)

[Como seu app vai tratar a persistência de dados?](#)

[Descreva qualquer caso de uso específico \("corner case"\) da experiência do Usuário \(UX\).](#)

[Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.](#)

[Descreva como você implementará o Google Play Services.](#)

## [Próximos Passos: Tarefas Necessárias](#)

[Tarefa 1: Configuração do Projeto](#)

[Tarefa 2: Registro e configuração do aplicativo](#)

[Tarefa 3: Criar base de dados no Firebase](#)

[Tarefa 4: Implementar camada de persistência](#)

[Tarefa 5: Implementar a Interface de Usuário \(UI\) para cada Activity e Fragment](#)

[Tarefa 6: Baixar listas de compras e itens padrão](#)

[Tarefa 7: Preferências](#)

[Tarefa 8: Criação de Lista de Compras](#)

[Tarefa 9: Modo Ir às Compras](#)

[Tarefa 10: Implementar o Widget](#)

**Usuário do GitHub:** nvolpini

# AppMercado

## Descrição

Aplicativo para gerenciar listas de compras de supermercado.

O usuário pode adicionar produtos, organizá-los por categoria, cadastrar preços por mercado e data e manter histórico das compras para comparação de preços.

No modo “Ir às Compras” o aplicativo pode apresentar ao usuário os mercados com melhores preços baseado no histórico de compras e enquanto estiver no mercado o usuário poderá marcar os itens que já foram adicionados ao carrinho de compras e controlar o valor total das compras.

Utilizando o serviço de localização o aplicativo identifica o mercado onde o usuário está fazendo compras automaticamente e destaca os itens com melhor preço naquele local, além de avisar quando o usuário estiver próximo a algum mercado com itens para comprar.

Através do widget é possível adicionar itens à lista de compras rapidamente sem precisar acessar o aplicativo. O widget permite selecionar uma lista de compras e vai mostrar os itens que não estão marcados para compra permitindo marcá-los para compra rapidamente.

O aplicativo permite fazer download de listas de compras padrão com itens populares para não precisar cadastrar todos os itens manualmente. Para utilizar essa funcionalidade o usuário deverá se cadastrar e autenticar (google ou e-mail/senha)

## Público-Alvo/Intended User

Qualquer pessoa que busque uma solução prática para controlar e organizar listas de compras de supermercado e facilitar a comparação de preços.

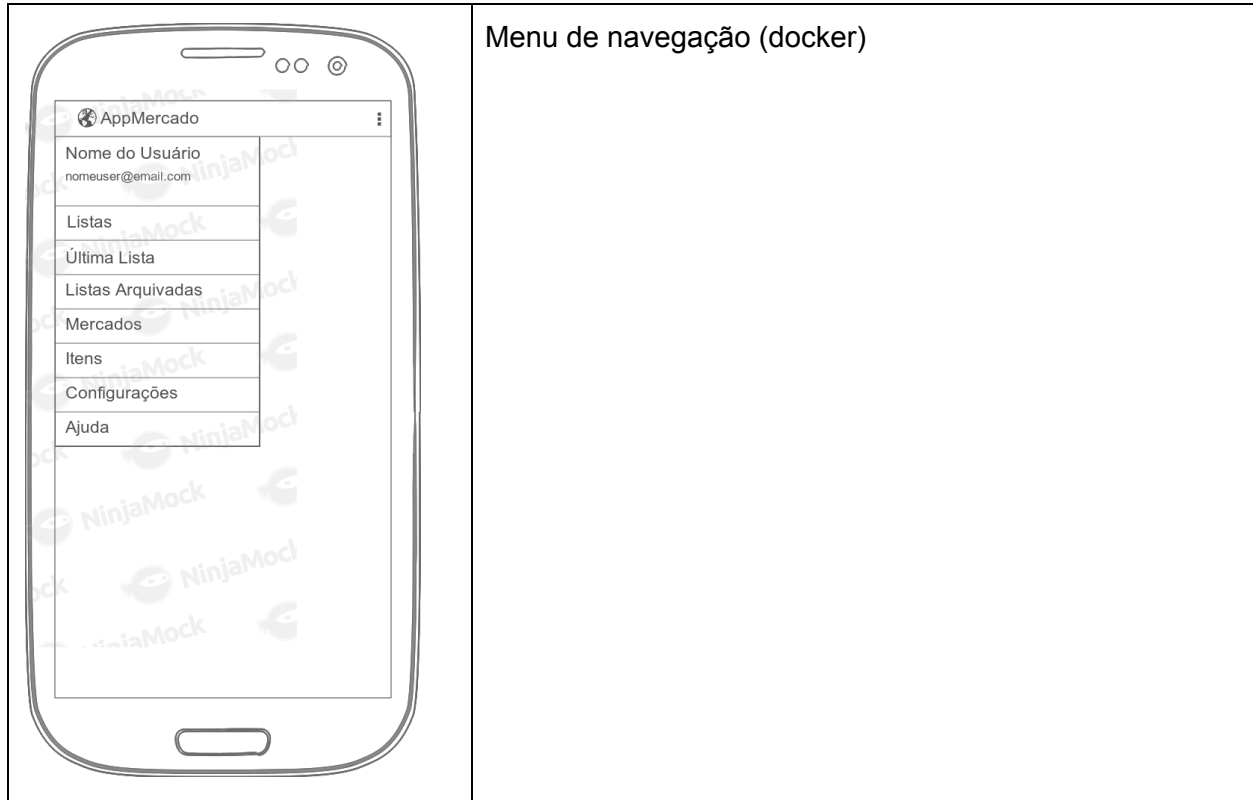
## Funcionalidades/Features

- Cadastrar itens
- Cadastrar categorias
- Cadastrar mercados
- Cadastrar preços
- Criar listas de compras
  - Adicionar itens
  - Remover itens
  - Marcar para compra
- Modo “Ir às Compras”
  - Identificar o mercado atual (permitir incluir mercado na hora da compra)
  - Listar itens a comprar e destacar preços
  - Cadastrar preços rapidamente
- Comparar preços

- Widget da lista de compras



## Protótipo de Interfaces do Usuário


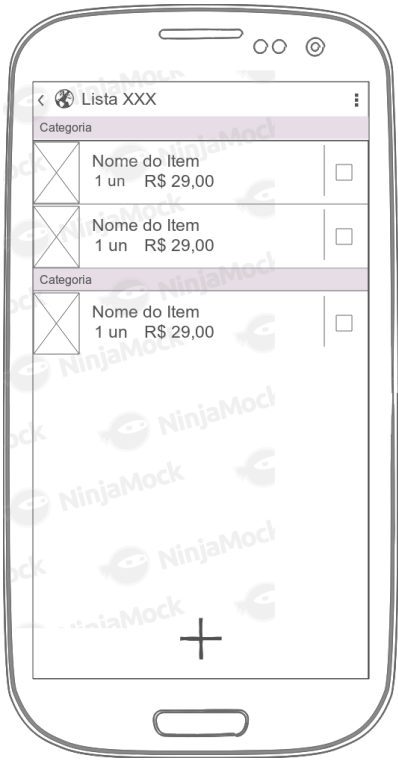
### Telas

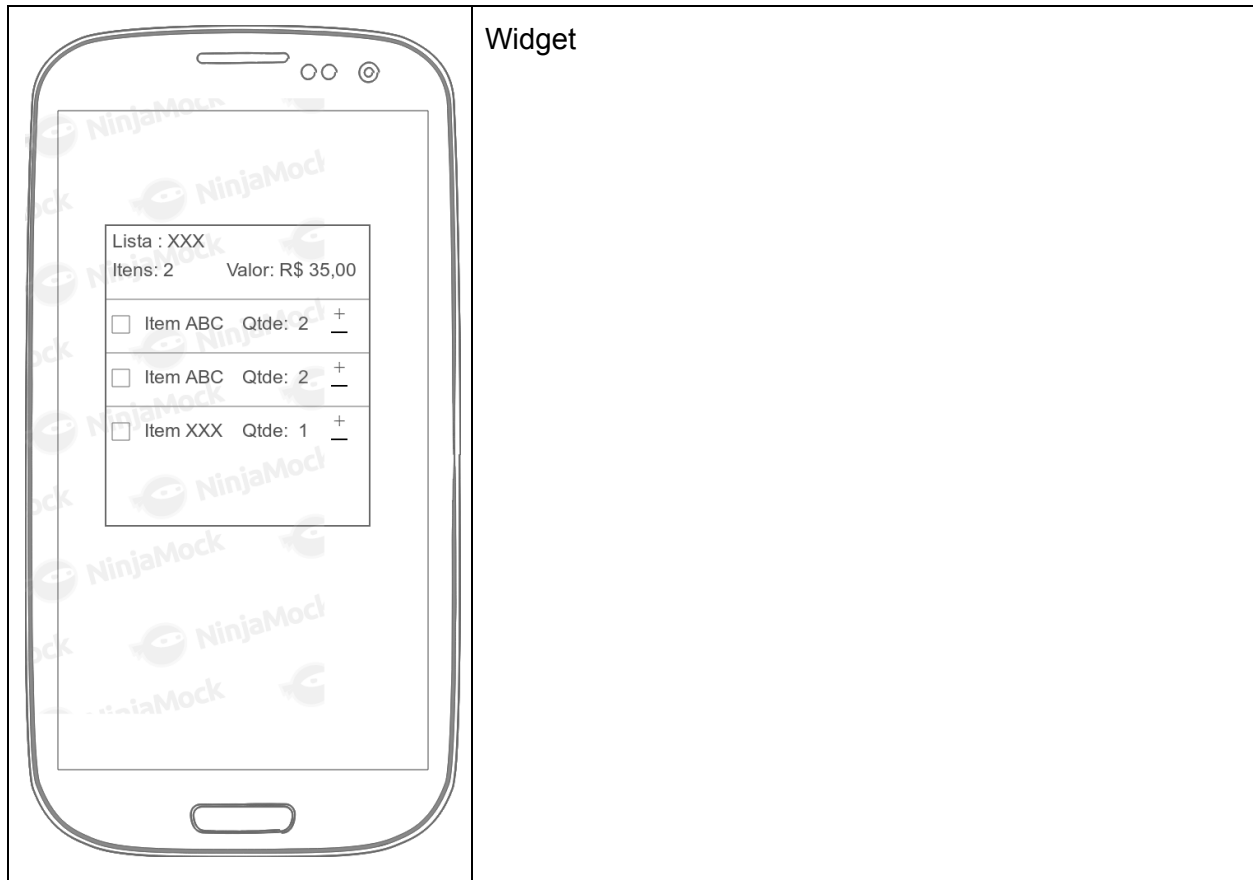


	<h2>Lista de mercados</h2>
	<h2>Cadastro de mercados - informações básicas</h2>

 <p>The mockup shows a mobile app interface for 'Cadastro de mercados - informações de localização'. At the top, there's a header bar with a back arrow, a globe icon, and the title 'Mercado'. Below the header, there's a tab bar with 'Dados' and 'Local'. The 'Local' tab is selected. The main content area includes a toggle switch for 'Habilitar Local' (currently OFF), two text input fields for 'Latitude' and 'Longitude', and three buttons: 'Informar Local', 'Remover Local', and 'Ver Local'.</p>	<h3>Cadastro de mercados - informações de localização</h3>
 <p>The mockup shows a mobile app interface for 'Cadastro de categorias'. At the top, there's a header bar with a back arrow, a globe icon, and the title 'Categoria'. Below the header, there's a text input field for 'Nome' and a toggle switch for 'Ativa' (currently ON). At the bottom, there's a button labeled 'Salvar' with a save icon.</p>	<h3>Cadastro de categorias</h3>

	<h2>Cadastro de itens</h2>
	<h2>Preços do item</h2>

	<h3>Tela das listas de compra cadastradas</h3>
	<h3>Detalhes da lista de compras</h3>



## Considerações Chave/Key Considerations

### Como seu app vai tratar a persistência de dados?

Os dados serão gravados em banco de dados SQLite e acessados através de ContentProvider.

SharedPreferences serão usadas para armazenar opções gerais do aplicativo.

Listas de compras padrão são disponibilizadas na Nuvem via Firebase.

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).



### Modo Ir as Compras

Ativando esse modo quando estiver no mercado o usuário pode “marcar” os itens que já foram comprados, restando na tela somente os itens ainda não comprados.

Com base na localização o aplicativo identifica o mercado onde o usuário se encontra e destaca se o preço naquele mercado é mais barato ou não.

O usuário pode cadastrar o preço rapidamente, basta clicar no item e informar o valor. O aplicativo vai gravar um novo preço para item naquela data e mercado. Os preços antigos são automaticamente arquivados.

**Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.**

Glide para carregar e fazer cache de imagens.

**Descreva como você implementará o Google Play Services.**

Location services e Mapas para identificar, mostrar e selecionar mercados.

Firebase para download de listas e itens comuns da Nuvem.

## Próximos Passos: Tarefas Necessárias

### Tarefa 1: Configuração do Projeto

Criar projeto no Android Studio.

Declarar as bibliotecas.

### Tarefa 2: Registro e configuração do aplicativo

Registrar o aplicativo no Developer Console do Google para ter acesso as API.

Registrar o aplicativo no Firebase Console.

### Tarefa 3: Criar base de dados no Firebase

Criar base de dados de listas e itens comuns no Firebase para fornecer dados inicial aos aplicativos.

## **Tarefa 4: Implementar camada de persistência**

Implementar banco de dados e ContentProvider.

Utilizar acpg para gerar código inicial para persistência.  
<https://github.com/BoD/android-contentprovider-generator>

## **Tarefa 5: Implementar a Interface de Usuário (UI) para cada Activity e Fragment**

- Activity/Fragment para Listar/Editar Mercados
- Activity/Fragment para Listar/Editar Unidades de Medida
- Activity/Fragment para Listar/Editar Categorias
- Activity/Fragment para Listar/Editar Itens
- Activity/Fragment para Listar/Editar Preços do Item
- Activity/Fragment para Listar/Editar Listas de Compras
- Activity/Fragment para Preferencias do aplicativo

## **Tarefa 6: Baixar listas de compras e itens padrão**

Implementar rotina para buscar itens padrão da Nuvem para criar uma base de dados inicial.

O usuário precisa se cadastrar para poder baixar. Pode logar com a conta do Google ou e-mail/senha criado na hora.

## **Tarefa 7: Preferências**

Habilitar Localização

- Para identificar mercados

Ultima lista

- Ao abrir o aplicativo, mostrar a última lista usada.
- Ao selecionar uma lista, gravar a preferência

## **Tarefa 8: Criação de Lista de Compras**

Adicionar itens

- Pesquisa por nome, por categorias, favoritos, compras recentes
- Criar novo item

Editar itens da lista

- Editar descrição, quantidade e unidade.
- Informar o preço e mercado
- Imagem (buscar de onde?)

Remover itens da lista

## **Tarefa 9: Modo Ir às Compras**

Identificar o mercado com base na localização.

Permitir que o usuário adicione/remova itens ao carrinho.

## **Tarefa 10: Implementar o Widget**

Ao adicionar o widget, permitir escolher a lista de compras.