

Projeto Integrador – Grupo V

App Shop em Flutter

Carlos Roberto do Val Jr.

Davi Macedo de Lima

João Luiz Marcelino Ribeiro

Thomaz Gabriel Azevedo de Oliveira

Pedro Henrique Sousa Avelar

Apresentação do Projeto

O projeto consiste no desenvolvimento de um aplicativo de e-commerce em Flutter, onde o aplicativo se comunica diretamente com o Firebase para gerenciar autenticação, armazenamento e atualização em tempo real dos dados. Essa abordagem garante uma experiência fluida e segura na compra online.

Descrição do Problema

- Justificativa / Contexto de Negócio:

O mercado de e-commerce exige soluções rápidas, interativas e seguras para atender a um público cada vez mais conectado. A necessidade de gerenciar um catálogo de produtos atualizado em tempo real, processar pedidos de forma ágil e oferecer uma experiência de usuário intuitiva justifica o desenvolvimento de um aplicativo mobile que se conecte diretamente ao Firebase, simplificando a integração entre o app e o banco de dados.

- Descrição das Personas:

- Consumidor Final: Usuário que busca uma experiência de compra intuitiva, com acesso rápido a um catálogo dinâmico, atualização em tempo real e funcionalidades como favoritos e histórico de pedidos.

- Administrador/Gestor: Responsável por gerenciar os produtos e acompanhar as métricas de vendas, podendo atualizar informações diretamente no banco de dados.

- Usuário Recorrente: Cliente fidelizado que valoriza uma experiência personalizada e o acompanhamento contínuo de seus pedidos por meio do aplicativo.

Macro Descrição da Proposta de Solução

- Principais Funcionalidades Previstas:

- Catálogo de produtos dinâmico com imagens, descrições e preços.

- Carrinho de compras interativo com atualização instantânea.

- Autenticação segura de usuários via Firebase (registro, login e logout).

- Processamento de pedidos e armazenamento de histórico de compras.

- Funcionalidade para marcar produtos como favoritos.

- Escopo do Projeto:

O aplicativo abrange:

- Frontend: Desenvolvimento da interface e experiência do usuário em Flutter, com gerenciamento de estado para garantir interatividade e atualizações em tempo real.
- Backend: Comunicação direta com o Firebase, que atuará como banco de dados e mecanismo de autenticação, eliminando a necessidade de um servidor intermediário.
- Integração e Segurança: Implementação de regras de negócio e segurança (HTTPS, criptografia de dados sensíveis) para proteger a comunicação entre o app e o Firebase.

Macro Cronograma:

- Sprint 1: Levantamento e definição dos requisitos, modelagem dos processos e elaboração dos diagramas (caso de uso, classes e MER/DER).
- Sprint 2: Configuração e integração com o Firebase, incluindo autenticação e armazenamento dos dados.
- Sprint 3: Desenvolvimento do aplicativo em Flutter e integração com o backend (Firebase).
- Sprint 4: Testes, ajustes, Quality Assurance (QA) e preparação para o lançamento.

Detalhamento da Proposta de Solução

1. Regras de Negócio

As regras de negócio definem como a aplicação deve se comportar para atender às demandas do negócio. Para um aplicativo de e-commerce, elas podem incluir:

- Validação de autenticação: Somente usuários autenticados podem adicionar itens ao carrinho ou realizar pedidos.
- Gerenciamento de estoque: Antes de adicionar um produto ao carrinho, o sistema deve verificar se há disponibilidade.
- Política de descontos: Aplicação de descontos somente se o valor do pedido ultrapassar um determinado limite ou se o usuário tiver um cupom válido.

- Processamento de pedidos: Uma vez finalizado o carrinho, o pedido deve ser criado, com associação de data, total e os itens comprados.
- Atualização em tempo real: As alterações no carrinho e na disponibilidade de produtos devem ocorrer de forma dinâmica, integrando com o Firebase em tempo real.

1.2 Requisitos Funcionais

Esses requisitos descrevem as funcionalidades que o sistema deve oferecer. Por exemplo:

- Catálogo de Produtos:
 - Exibir uma lista de produtos com imagem, descrição e preço.
- Permitir a pesquisa e filtragem por categorias.
- Carrinho de Compras:
 - Adicionar produtos ao carrinho.
 - Atualizar quantidades e remover itens.
- Autenticação de Usuários:
 - Cadastro, login, logout e autenticação automática usando Firebase.
- Processamento de Pedidos:
 - Gerar um pedido a partir do carrinho, salvar data, total e itens do pedido.
 - Exibir histórico de pedidos ao usuário.
- Favoritos:
 - Permitir marcar produtos como favoritos e filtrá-los.
- Formulário Dinâmico:
 - Receber um link e gerar automaticamente uma imagem para o produto, armazenando as informações no Firebase.

1.3 Requisitos Não Funcionais

Esses requisitos definem “como” o sistema deve funcionar em termos de qualidade e desempenho:

- Desempenho:
 - Atualização em tempo real do carrinho e catálogo com resposta rápida.
- Segurança:
 - Comunicação segura (HTTPS), criptografia dos dados sensíveis e proteção de token de autenticação.
- Usabilidade:
 - Interface intuitiva, com animações que facilitam a navegação e uma experiência agradável para o usuário.
- Escalabilidade:
 - Arquitetura capaz de suportar aumento no número de usuários e transações, principalmente pela integração com Firebase.
- Manutenibilidade:
 - Código modular (usando MobX e Providers, por exemplo) para facilitar atualizações e correções futuras.

2. Diagramas

2.1 Diagrama de Caso de Uso

Esse diagrama ajuda a visualizar as interações entre os atores e as funcionalidades do sistema. Por exemplo, os principais atores podem ser:

- Usuário Comum:
 - Visualizar catálogo de produtos.
 - Adicionar/Remover itens do carrinho.
 - Finalizar pedido.
 - Visualizar histórico de pedidos.
 - Marcar produtos como favoritos.
- Administrador (opcional):

Gerenciar (criar, atualizar, excluir) produtos. • Gerenciar pedidos e visualizar relatórios de vendas.

Você pode representar esses casos de uso em um diagrama simples, com elipses para cada funcionalidade e conectando-os aos respectivos atores (ícones ou stick figures).

2.2 Diagrama de Classes

O diagrama de classes ilustra a estrutura do sistema. Com base no código fornecido, algumas classes essenciais são:

- Auth: Gerencia autenticação, tokens e logout automático.
- Cart e CartItem: Representam o carrinho de compras e seus itens.
- Order e OrderList: Tratam a criação e gerenciamento de pedidos.
- Product e ProductList: Gerenciam os produtos, incluindo a funcionalidade de favoritar.

O diagrama deve mostrar as associações entre essas classes (por exemplo, o Cart possui muitos CartItem; o Order contém uma lista de CartItem e assim por diante).

3. Modelos e Banco de Dados

3.1 Modelo Entidade Relacionamento (MER/DER)

Para o banco de dados do seu e-commerce, você pode definir entidades como:

- Usuário:
 - Atributos: id, nome, email, senha, etc.
- Produto:
 - Atributos: id, nome, descrição, preço, imageUrl, isFavorite.
- Pedido:
 - Atributos: id, data, total, idUsuario (chave estrangeira).
- ItemPedido (ou Carrinho):
 - Atributos: id, idPedido (FK), idProduto (FK), quantidade, preço.

- Favoritos (opcional): • Relação entre Usuário e Produto para marcar favoritos.

Modelo Entidade-Relacionamento (MER):

1. Entidades e Atributos:

a. Usuário:

- i. ID_Usuário (chave primária)
- ii. Nome
- iii. Email
- iv. Senha
- v. Endereço
- vi. Telefone
- vii. Data_de_Registro

b. Produto:

- i. ID_Produto (chave primária)
- ii. Nome
- iii. Descrição
- iv. Preço
- v. Estoque
- vi. Categoria
- vii. Imagem_URL

c. Pedido:

- i. ID_Pedido (chave primária)
- ii. Data_do_Pedido
- iii. Status
- iv. ID_Usuário (chave estrangeira)

d. Item_Pedido:

- i. ID_Item_Pedido (chave primária)
- ii. Quantidade
- iii. Preço_Total
- iv. ID_Pedido (chave estrangeira)
- v. ID_Produto (chave estrangeira)

e. Categoria:

- i. ID_Categoria (chave primária)
- ii. Nome
- iii. Descrição

f. Favorito:

- i. ID_Favorito (chave primária)
- ii. ID_Usuário (chave estrangeira)
- iii. ID_Produto (chave estrangeira)

2. Relacionamentos:

- a. **Usuário** realiza **Pedido**: relação de 1:N (um usuário pode realizar múltiplos pedidos).
- b. **Pedido** contém **Item_Pedido**: relação de 1:N (um pedido pode conter múltiplos itens).
- c. **Item_Pedido** refere-se a **Produto**: relação de N:1 (muitos itens podem referir-se a um produto).
- d. **Produto** pertence a **Categoria**: relação de N:1 (muitos produtos podem pertencer a uma categoria).
- e. **Usuário** tem **Favorito**: relação de N:M (um usuário pode ter múltiplos produtos favoritos e um produto pode ser favorito de múltiplos usuários).

Diagrama Entidade-Relacionamento (DER):

Para visualizar o DER, considere as seguintes representações:

- **Entidades**: representadas por retângulos.
- **Atributos**: representados por elipses conectadas às suas respectivas entidades.
- **Relacionamentos**: representados por losangos conectando as entidades envolvidas.

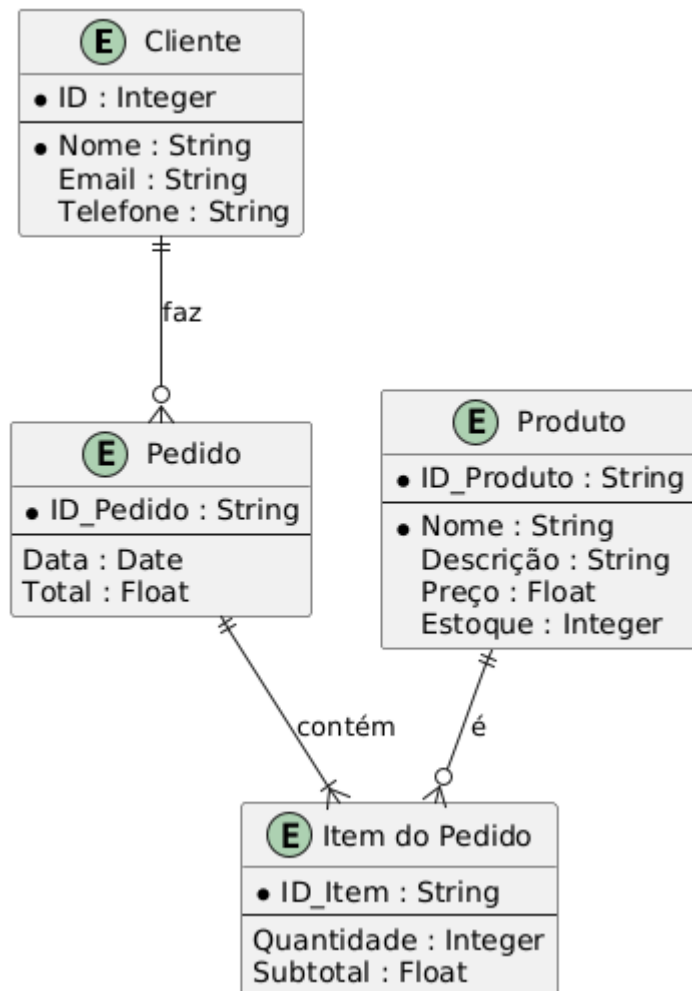
Descrição do DER:

- A entidade **Usuário** possui os atributos ID_Usuário, Nome, Email, Senha, Endereço, Telefone e Data_de_Registro.
- A entidade **Produto** possui os atributos ID_Produto, Nome, Descrição, Preço, Estoque, Categoria e Imagem_URL.
- A entidade **Pedido** possui os atributos ID_Pedido, Data_do_Pedido, Status e ID_Usuário.
- A entidade **Item_Pedido** possui os atributos ID_Item_Pedido, Quantidade, Preço_Total, ID_Pedido e ID_Produto.
- A entidade **Categoria** possui os atributos ID_Categoria, Nome e Descrição.
- A entidade **Favorito** possui os atributos ID_Favorito, ID_Usuário e ID_Produto.

Relacionamentos:

- **Usuário** — (realiza) — **Pedido**
- **Pedido** — (contém) — **Item_Pedido**
- **Item_Pedido** — (refere-se a) — **Produto**
- **Produto** — (pertence a) — **Categoria**
- **Usuário** — (tem) — **Favorito** — (de) — **Produto**

DER – Imagem



Dicionário de Dados:

1. Usuário:

- a. **Descrição:** Representa os clientes que utilizam o aplicativo para realizar compras.
- b. **Atributos:**
 - i. **ID_Usuário:** Identificador único do usuário.
 1. *Tipo:* String (UID gerado pelo Firebase Authentication).

- 2. *Restrições*: Chave primária, não nulo, único.
- ii. Nome: Nome completo do usuário.
 - 1. *Tipo*: String.
 - 2. *Restrições*: Não nulo.
- iii. Email: Endereço de e-mail do usuário.
 - 1. *Tipo*: String.
 - 2. *Restrições*: Não nulo, único, formato de e-mail válido.
- iv. Senha: Senha para autenticação.
 - 1. *Tipo*: String.
 - 2. *Restrições*: Não nulo, armazenada de forma criptografada.
- v. Endereço: Localização física do usuário para entregas.
 - 1. *Tipo*: String.
 - 2. *Restrições*: Opcional.
- vi. Telefone: Número de contato do usuário.
 - 1. *Tipo*: String.
 - 2. *Restrições*: Opcional, formato de telefone válido.
- vii. Data_de_Registro: Data em que o usuário se registrou no aplicativo.
 - 1. *Tipo*: Timestamp.
 - 2. *Restrições*: Não nulo, valor padrão é a data e hora do registro.

2. Produto:

- a. **Descrição**: Itens disponíveis para compra no aplicativo.
- b. **Atributos**:
 - i. ID_Produto: Identificador único do produto.
 - 1. *Tipo*: String (gerado automaticamente).
 - 2. *Restrições*: Chave primária, não nulo, único.
 - ii. Nome: Nome do produto.
 - 1. *Tipo*: String.
 - 2. *Restrições*: Não nulo.
 - iii. Descrição: Detalhes sobre o produto.
 - 1. *Tipo*: String.
 - 2. *Restrições*: Opcional.
 - iv. Preço: Custo unitário do produto.
 - 1. *Tipo*: Decimal.
 - 2. *Restrições*: Não nulo, maior que zero.
 - v. Estoque: Quantidade disponível em estoque.
 - 1. *Tipo*: Inteiro.
 - 2. *Restrições*: Não nulo, zero ou positivo.
 - vi. Categoria: Categoria à qual o produto pertence.
 - 1. *Tipo*: String (referência ao ID_Categoria).

- 2. *Restrições*: Não nulo, chave estrangeira.
- vii. Imagem_URL: Link para a imagem do produto.
 - 1. *Tipo*: String (URL).
 - 2. *Restrições*: Opcional, formato de URL válido.

3. Pedido:

- a. **Descrição**: Registra as compras realizadas pelos usuários. [Firebase](#)
- b. **Atributos**:
 - i. ID_Pedido: Identificador único do pedido.
 - 1. *Tipo*: String (gerado automaticamente).
 - 2. *Restrições*: Chave primária, não nulo, único.
 - ii. Data_do_Pedido: Data e hora em que o pedido foi realizado.
 - 1. *Tipo*: Timestamp.
 - 2. *Restrições*: Não nulo.
 - iii. Status: Estado atual do pedido (e.g., "Pendente", "Enviado", "Entregue").
 - 1. *Tipo*: String.
 - 2. *Restrições*: Não nulo, valores pré-definidos.
 - iv. ID_Usuário: Identificador do usuário que realizou o pedido.
 - 1. *Tipo*: String (referência ao ID_Usuário).
 - 2. *Restrições*: Não nulo, chave estrangeira.

4. Item_Pedido:

- a. **Descrição**: Itens individuais que compõem um pedido.
- b. **Atributos**:
 - i. ID_Item_Pedido: Identificador único do item do pedido.
 - 1. *Tipo*: String (gerado automaticamente).
 - 2. *Restrições*: Chave primária, não nulo, único.

Diagrama de caso de Uso:

