

Projeto Prático (P2)

Objetivo

Desenvolver um aplicativo mobile utilizando o framework Flutter, que **permita aos usuários (clientes) visualizar o cardápio de um restaurante e realizar pedidos**. O foco do projeto será a implementação de uma interface simples e funcional, que permita a navegação entre as seções do cardápio e a adição de itens a um pedido. As funcionalidades deverão ser integradas à plataforma de serviços Google Firebase.

Requisitos Funcionais

RF001	Autenticação, Criação de Conta e Recuperação de Senha
	<p>O aplicativo deve integrar suas funcionalidades de Login, Criação de Conta e Recuperação de Senha ao serviço <i>Firebase Authentication</i>.</p> <ul style="list-style-type: none">● Autenticação (Login): deve garantir que apenas usuários previamente cadastrados tenham acesso às funcionalidades do aplicativos. As mensagens de erro, tais como, formato de email inválido, credenciais de acesso inválidas, entre outras devem ser devidamente tratadas e exibidas para o usuário.● Criação de Conta: Para criação de novas contas, o usuário deve fornecer as seguintes informações: nome, e-mail, senha e confirmação de senha (outras informações também podem ser inseridas caso seja necessário). Os dados a respeito do e-mail e senha serão gerenciados pelo <i>Firebase Authentication</i>, enquanto, o nome deve ser armazenado no <i>Firebase Firestore</i> em uma coleção denominada <i>usuarios</i>.● Recuperação de senha: Utilizar os recursos do <i>Firebase Authentication</i> para enviar o email de recuperação de senha.

RF002 Categorias e Itens do Cardápio

Todas as informações utilizadas para visualização do cardápio, bem como os detalhes dos itens deverão ser armazenadas no serviço *Firebase Firestore*.

Para armazenar as informações a respeito das **categorias**, a coleção deverá utilizar a seguinte estrutura:

Coleção: categorias	
Campo	Descrição
nome	Nome da categoria
descrição	Breve descrição da categoria
imagem	URL ou nome da imagem representativa (opcional)
ordem	Número para ordenação no menu (opcional)

Exemplo:

```
{
  "nome": "Prato Principal",
  "descricao": "Pratos principais servidos",
  "imagem": "link-da-imagem",
  "ordem": 2
}
```

Os itens que fazem parte do cardápio serão armazenados na coleção **itens_cardapio**, considerando a seguinte estrutura:

Coleção: itens_cardapio	
Campo	Descrição
nome	nome do item do cardápio
descricao	Descrição detalhada do item, incluindo informações sobre ingredientes, preparo, características especiais e quais outras informações relevantes
preco	Preço do item
imagem	URL ou nome da imagem do item
ativo	Status da disponibilidade do item (true ou false)
categoria	ID da categoria ou nome da categoria associada a este item

Exemplo:

```
{
  "nome": "Filé com Fritas",
  "descricao": "Filé grelhado com batatas fritas",
  "preco": 45.00,
  "imagem": "link-da-imagem-file",
  "ativo" : true,
  "categoria": "Prato Principal",
}
```

RF003**Pedido**

Ao selecionar um item no cardápio, o usuário deverá ser redirecionado para uma tela que exibe informações detalhadas sobre o item escolhido. O botão "Adicionar ao Pedido" faz com que o item escolhido seja adicionado a uma coleção no *Firebase Firestore* denominada **pedidos**.

Coleção: pedidos	
Campo	Descrição
uid	UID do Cliente que está realizando o pedido
status	Indica do status do pedido: {preparando ou finalizado}
data_hora	Data e hora do pedido
itens	Subcoleção que armazena os itens que fazem parte do pedido, incluindo <i>ID do item</i> , o <i>preço</i> e a <i>quantidade</i> .

Exemplo:

```
{
  "uid": "ID do cliente",
  "status": "preparando",
  "data_hora": "01/01/2001 13:10"
  "itens": [
    {"item_id": "idItem1", "preco": 45.00, "quantidade": 2},
    {"item_id": "idItem2", "preco": 20.00, "quantidade": 1}
  ],
}
```

Na interface de *Resumo do Pedido* o usuário poderá:

- Alterar as quantidades ou remover itens do pedido
- Visualizar o total geral do pedido
- Confirmar o pedido, em que o status deverá ser alterado de preparando para finalizado. Quando esta ação ocorrer, os itens que fazem parte do resumo do pedido não deverão ser mais exibidos.

Requisitos Não-Funcionais

- O aplicativo deve ser desenvolvido com o Flutter SDK.
- O aplicativo deve ser integrado aos serviços da plataforma Firebase
- O aplicativo poderá ser desenvolvido individualmente ou em duplas.

O que deverá ser entregue?

- A entrega será realizada através de um repositório público no GitHub. Para isso, o desenvolvedor deverá criar um repositório e hospedar o código fonte completo do aplicativo. Entregar um arquivo texto contendo:
 - nomes e códigos dos alunos envolvidos
 - endereço do repositório do GitHub
 - endereço do aplicativo publicado no Firebase Hosting
- 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 Autenticação, Criação de Conta e Recuperação de Senha	Pontuação
<ul style="list-style-type: none"><input type="checkbox"/> Autenticação integrada ao Firebase Authentication.<input type="checkbox"/> Criação de Conta integrada ao Firebase Authentication.<input type="checkbox"/> Recuperação de senha integrada ao Firebase Authentication.	
[40%] RF002 Categorias e Itens do Cardápio	Pontuação
<ul style="list-style-type: none"><input type="checkbox"/> Os dados das categorias são armazenados na respectiva coleção e exibidos adequadamente no aplicativo.<input type="checkbox"/> Os dados dos itens do cardápio são armazenados na respectiva coleção.<input type="checkbox"/> A visualização do cardápio é realizada dinamicamente a partir das coleções <i>categorias</i> e <i>itens_cardapio</i>.	
[50%] RF003 Pedido	Pontuação
<ul style="list-style-type: none"><input type="checkbox"/> Os itens do cardápio são devidamente adicionados à coleção de <i>pedidos</i>.<input type="checkbox"/> As alterações dos itens que fazem parte do pedido são atualizadas no banco de dados.<input type="checkbox"/> As remoções de itens que fazem parte do pedido são replicadas no banco de dados.<input type="checkbox"/> O cálculo do valor total é realizado corretamente e exibido para o usuário.<input type="checkbox"/> Ao confirmar o pedido, o status é alterado para finalizado.	

Considerações Finais

- A identificação de cópias indiscriminadas de código fonte da Internet, 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.