

# Plano de Testes - Aplicativo GastroMatch

## 1 .Visão geral

A aplicação GastroMatch foi submetida a uma bateria de testes unitários com o objetivo de garantir a robustez, usabilidade e confiabilidade das funcionalidades principais oferecidas aos usuários, tanto clientes quanto chefs.

Os testes desenvolvidos abordaram as seguintes páginas e componentes críticos:

- **ChefDetailPage**
- **ChefSearch**
- **EditReservationPage**
- **LoginPage**
- **UserProfileEditPage**
- **UserReservationsPage**
- **UserPastReservationsPage**

Esses testes validaram os principais fluxos de interação, como:

- Preenchimento de formulários
- Envio de dados e tratamento de erros
- Navegação entre telas
- Comportamento de carregamento e exibição de mensagens

## 2. Estratégia de Teste

A abordagem de testes foi definida com base na estrutura da pasta 'lib'. Os testes foram divididos entre unitários (para validação de lógica de negócio e models) e de widget (para validar interfaces).

## Cobertura Esperada

O objetivo inicial foi atingir pelo menos **60% de cobertura de código** da camada de apresentação (UI), que representa o núcleo das interações da aplicação.

## Cobertura Realizada

A cobertura obtida com os testes realizados atingiu aproximadamente **67% do código das páginas testadas**, segundo a análise da ferramenta de cobertura utilizada via `flutter test --coverage`.

## 3. Tipos de Testes Aplicados

### 3.1 Como executar os testes

```
cd code/mobile/gastromatch_mobile  
  
flutter pub get  
  
flutter test
```

### 3.2 Testes Unitários

- Modelo [Reservation](#):
  - Verificação de criação correta via método *fromJson*.
  - Validação de atributos esperados após *parse* do *JSON*.
- Página [ChefSearch](#):
  - Filtro correto ao digitar nome de chef.
  - Navegação para a página de detalhes do chef ao clicar.
- Página [EditReservationPage](#):
  - Campos preenchidos corretamente com dados da reserva.
  - Validação da ausência de data/hora.
  - Envio dos dados ao backend com token JWT.
- Página [LoginPage](#):
  - Exibição de campos de email e senha.
  - Indicador de carregamento ao fazer login.
  - Exibição de erro com credenciais inválidas.
  - Navegação para página de cadastro.

- Página [UserProfileEditPage](#):
  - Carregamento dos dados do usuário.
  - Edição e envio dos dados via Multipart PATCH.
  - Validação visual da foto de perfil e feedback por snackbar.
  
- Página [UserReservationsPage](#):
  - Exibição de carregamento, erro ou lista vazia.
  - Listagem das reservas com informações.
  - Exclusão de reservas com confirmação por AlertDialog.
  
- Página [UserPastReservationsPage](#):
  - Busca de reservas anteriores do usuário.
  - Busca do nome do chef vinculado.
  - Exibição de botão de avaliação caso sem nota.