



Instruções de Setup Completo

1. Configuração Inicial do Projeto

Criar Projeto Flutter

```
flutter create delivery_app_cliente  
cd delivery_app_cliente
```

Substituir arquivos

1. Substitua `pubspec.yaml` pelo fornecido
2. Execute: `flutter pub get`
3. Copie toda a estrutura de pastas `lib/`
4. Copie arquivos da pasta `web/`
5. Copie `analysis_options.yaml`
6. Copie `firebase.json`
7. Copie `firestore.rules`

2. Configuração Firebase

Passo 1: Criar Projeto Firebase

1. Acesse [Firebase Console](https://console.firebase.google.com) (<https://console.firebase.google.com>)
2. Crie novo projeto
3. Ative Authentication, Firestore, Functions, Hosting, Cloud Messaging
4. Configure domínio para PWA

Passo 2: Configurar FlutterFire

```
# Instalar Firebase CLI  
npm install -g firebase-tools  
  
# Login no Firebase  
firebase login  
  
# Instalar FlutterFire CLI  
dart pub global activate flutterfire_cli  
  
# Configurar projeto  
flutterfire configure
```

Passo 3: Substituir `firebase_options.dart`

- O comando acima gera `firebase_options.dart`
- Substitua pelo arquivo fornecido e complete com suas chaves

Passo 4: Configurar Authentication

1. No Firebase Console → Authentication → Sign-in method

2. Ative “Email/password”
3. Ative “Google” e configure OAuth

Passo 5: Configurar Firestore

1. Substitua as regras pelo arquivo `firestore.rules`
2. Crie índices necessários
3. Popule dados iniciais (opcional)

Passo 6: Configurar Cloud Functions

```
cd functions
npm install
npm run build
firebase deploy --only functions
```

Passo 7: Configurar Cloud Messaging

1. Gere chave VAPID no Firebase
2. Substitua ‘YOUR_VAPID_KEY’ no código
3. Configure service worker

3. Configuração MercadoPago

Criar Conta MercadoPago

1. Acesse [MercadoPago Developers](https://www.mercadopago.com.br/developers) (<https://www.mercadopago.com.br/developers>)
2. Crie aplicação
3. Obtenha chaves de sandbox e produção

Configurar Chaves

```
# No Firebase Functions
firebase functions:config:set mercadopago.access_token="YOUR_ACCESS_TOKEN"
firebase functions:config:set mercadopago.public_key="YOUR_PUBLIC_KEY"

# Deploy functions
firebase deploy --only functions
```

Substituir Chaves no Código

- `web/index.html` : `YOUR_MERCADOPAGO_PUBLIC_KEY`
- `functions/src/index.ts` : Chaves já são carregadas via config

4. Dados Iniciais Firestore

Estrutura de Coleções

restaurants

```
{
  "name": "Pizzaria Italiana",
  "description": "Autênticas pizzas italianas",
  "imageUrl": "https://upload.wikimedia.org/wikipedia/commons/c/c8/Pizza_Margherita_stu_spivack.jpg",
  "category": "Italiana",
  "rating": 4.5,
  "reviewCount": 150,
  "deliveryFee": 5.0,
  "deliveryTime": 35,
  "isOpen": true,
  "isFeatured": true,
  "tags": ["pizza", "italiana", "massa"],
  "address": {
    "street": "Rua das Flores",
    "number": "123",
    "neighborhood": "Centro",
    "city": "São Paulo",
    "state": "SP",
    "zipCode": "01234-567",
    "latitude": -23.5505,
    "longitude": -46.6333
  },
  "workingHours": {
    "monday": { "isOpen": true, "openTime": "18:00", "closeTime": "23:00" },
    "tuesday": { "isOpen": true, "openTime": "18:00", "closeTime": "23:00" },
    "wednesday": { "isOpen": true, "openTime": "18:00", "closeTime": "23:00" },
    "thursday": { "isOpen": true, "openTime": "18:00", "closeTime": "23:00" },
    "friday": { "isOpen": true, "openTime": "18:00", "closeTime": "00:00" },
    "saturday": { "isOpen": true, "openTime": "18:00", "closeTime": "00:00" },
    "sunday": { "isOpen": true, "openTime": "18:00", "closeTime": "23:00" }
  }
}
```

categories

```
{
  "name": "Pizzas",
  "description": "Deliciosas pizzas artesanais",
  "imageUrl": "https://i.ytimg.com/vi/UYVorC1l7V4/hqdefault.jpg",
  "order": 1
}
```

products

```
{
  "name": "Pizza Margherita",
  "description": "Molho de tomate, mussarela, manjericão fresco",
  "imageUrl": "https://i.ytimg.com/vi/--bv0V6ZjWI/sddefault.jpg",
  "price": 35.90,
  "categoryId": "category_id_here",
  "restaurantId": "restaurant_id_here",
  "isAvailable": true,
  "preparationTime": 25,
  "allergens": ["glúten", "lactose"],
  "options": [
    {
      "id": "size",
      "name": "Tamanho",
      "isRequired": true,
      "items": [
        { "id": "small", "name": "Pequena", "additionalPrice": 0 },
        { "id": "medium", "name": "Média", "additionalPrice": 5 },
        { "id": "large", "name": "Grande", "additionalPrice": 10 }
      ]
    }
  ],
  "addons": [
    {
      "id": "extra_cheese",
      "name": "Queijo Extra",
      "price": 4.50,
      "description": "Mussarela adicional",
      "isAvailable": true
    }
  ]
}
```

5. Build e Deploy

Build para Web

```
flutter build web --web-renderer html --release
```

Deploy no Firebase

```
firebase deploy
```

PWA Features

- ☒ Instalável
- ☒ Offline fallback
- ☒ Service Worker
- ☒ Manifest configurado
- ☒ Ícones PWA

6. Desenvolvimento Local

Executar em desenvolvimento

```
flutter run -d chrome --web-port 3000
```

Executar Firebase Emulators

```
firebase emulators:start
```

7. Monitoramento

Firebase Analytics

- Eventos automáticos configurados
- Métricas de conversão
- Funil de compra

Crashlytics

- Relatórios automáticos de crashes
- Logs personalizados
- Métricas de estabilidade

8. Próximos Passos

1. ☒ Configure as chaves do Firebase
2. ☒ Configure as chaves do MercadoPago
3. ☒ Popule dados iniciais no Firestore
4. ☒ Teste o fluxo completo
5. ☒ Configure domínio personalizado
6. ☒ Configure CI/CD (opcional)

9. Troubleshooting

Problemas Comuns

- **CORS:** Configure CORS no Firebase Hosting
- **PWA não instala:** Verifique manifest.json e service worker
- **Notificações não funcionam:** Verifique chave VAPID
- **Pagamentos falham:** Verifique chaves MercadoPago

Logs úteis

```
# Firebase Functions logs
firebase functions:log

# Flutter web logs
flutter run -d chrome --web-port 3000 --verbose
```



Seu app cliente está configurado e pronto para uso. Agora você pode:

- Fazer login/cadastro
- Navegar por restaurantes
- Adicionar itens ao carrinho
- Finalizar pedidos
- Acompanhar entregas
- Receber notificações

Para App Restaurante e Painel Admin, solicite em sessões separadas!