

# Guia de Deployment - FutPerform Club Manager

---

Este guia explica como fazer o deployment completo da aplicação FutPerform Club Manager no Firebase.

## Pré-requisitos

---

- Node.js 18+ instalado
- Firebase CLI instalado globalmente ( `npm install -g firebase-tools` )
- Projeto Firebase criado ([console.firebase.google.com](https://console.firebase.google.com) (<https://console.firebase.google.com>))
- Credenciais Firebase configuradas em `nextjs_space/lib/firebase.js`

## Configuração Inicial

---

### 1. Ativar Serviços Firebase

No [Firebase Console](https://console.firebase.google.com) (<https://console.firebase.google.com>), ative:

1. **Authentication**
  - Email/Password
  - Google Sign-In
2. **Firestore Database**
  - Modo de produção
  - Região: europe-west1 (recomendado para Portugal)
3. **Cloud Functions**
  - Upgrade para plano Blaze (pay-as-you-go)
4. **Hosting** (opcional)
  - Para deploy do frontend
5. **Storage**
  - Para fotos de perfil e documentos

## 2. Configurar Firebase CLI

```
# Login no Firebase
firebase login

# Ir para o diretório raiz do projeto
cd /home/ubuntu/futperform_club_manager

# Inicializar Firebase
firebase init

# Selecionar:
# - Firestore (regras e índices)
# - Functions (JavaScript)
# - Hosting (opcional)

# Seguir prompts:
# - Projeto: futperform-club-manager
# - Firestore rules: firestore.rules
# - Firestore indexes: firestore.indexes.json
# - Functions: JavaScript (não TypeScript)
# - Functions diretório: functions
# - Hosting public: nextjs_space/out
```



## Deploy das Cloud Functions

### 1. Instalar Dependências

```
cd functions
npm install
```

### 2. Testar Localmente (Emulators)

```
# Voltar ao diretório raiz
cd ..

# Iniciar emuladores
firebase emulators:start

# Aceder Firebase UI: http://localhost:4000
# - Functions: http://localhost:5001
# - Firestore: http://localhost:8080
# - Auth: http://localhost:9099
```

### 3. Deploy para Produção

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

# OU deploy completo
firebase deploy
```

## 4. Verificar Deploy

```
# Ver logs das functions
firebase functions:log

# Listar functions deployadas
firebase functions:list
```



## Deploy das Regras Firestore

```
# Deploy apenas das regras
firebase deploy --only firestore:rules

# Deploy dos índices
firebase deploy --only firestore:indexes
```



## Deploy do Frontend (Next.js)

### Opção 1: Firebase Hosting

```
cd nextjs_space

# Build estático
yarn build
yarn export

# Deploy
cd ..
firebase deploy --only hosting
```

### Opção 2: Vercel (Recomendado para Next.js)

```
# Instalar Vercel CLI
npm install -g vercel

cd nextjs_space

# Deploy
vercel

# Seguir prompts e configurar variáveis de ambiente
```

### Opção 3: Export Estático

```
cd nextjs_space

# Build
yarn build

# Ficheiros gerados em .next/
# Fazer upload para hosting de escolha
```

## Configurar Custom Claims

As Cloud Functions automaticamente configuram custom claims ao adicionar membros.

Para configurar manualmente:

```
// No Firebase Console → Authentication → Users
// Ou via Admin SDK:

const admin = require('firebase-admin');
admin.initializeApp();

await admin.auth().setCustomUserClaims('USER_UID', {
  role: ['Presidente'],
  departments: ['Direção'],
  clubId: 'default-club'
});
```

## Configurar Notificações por Email

### Opção 1: Firebase Extensions (SendGrid)

```
# Instalar extensão SendGrid
firebase ext:install sendgrid/sendgrid-email-delivery

# Seguir prompts para configurar API Key
```

### Opção 2: Configuração Manual

1. Criar conta [SendGrid](https://sendgrid.com) (<https://sendgrid.com>)
2. Gerar API Key
3. Configurar environment variable:

```
firebase functions:config:set sendgrid.key="SG.xxxxxxxxxxxxxx"

# Verificar configuração
firebase functions:config:get
```

1. Atualizar código das functions para enviar emails:

```
const sgMail = require('@sendgrid/mail');
sgMail.setApiKey(functions.config().sendgrid.key);

// Enviar email
await sgMail.send({
  to: 'destinatario@exemplo.pt',
  from: 'noreply@futperformclub.pt',
  subject: 'Nova Requisição',
  text: 'Mensagem...',
  html: '<strong>Mensagem...</strong>'
});
```

## Testar em Produção

### 1. Criar Utilizadores de Teste

```
// Via Firebase Console ou Admin SDK
const auth = admin.auth();

// Criar utilizador
const user = await auth.createUser({
  email: 'teste@clube.pt',
  password: 'senha123',
  displayName: 'Utilizador Teste'
});

// Adicionar custom claims
await auth.setCustomUserClaims(user.uid, {
  role: ['Treinador'],
  departments: ['Equipa Técnica'],
  clubId: 'default-club'
});
```

### 2. Popular Dados de Teste

```
# Executar seed script
node scripts/seed-data.js
```

### 3. Testar Funcionalidades

#### 1. Login/Autenticação

- Email/Password
- Google Sign-In

#### 2. Departamentos

- Criar departamento
- Adicionar membros
- Atualizar disponibilidade

#### 3. Requisições

- Enviar requisição entre departamentos
- Marcar como concluída

#### 4. Jogadores

- Adicionar jogador
- Adicionar observação médica/técnica

#### 5. Feed

- Criar post
- Verificar visibilidade por departamento

## Continuous Deployment (CI/CD)

### GitHub Actions

Criar `.github/workflows/deploy.yml` :

```

name: Deploy to Firebase

on:
  push:
    branches: [ main ]

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Install dependencies (Functions)
        run: |
          cd functions
          npm install

      - name: Install dependencies (Frontend)
        run: |
          cd nextjs_space
          yarn install

      - name: Build Next.js
        run: |
          cd nextjs_space
          yarn build

      - name: Deploy to Firebase
        uses: FirebaseExtended/action-hosting-deploy@v0
        with:
          repoToken: '${{ secrets.GITHUB_TOKEN }}'
          firebaseServiceAccount: '${{ secrets.FIREBASE_SERVICE_ACCOUNT }}'
          projectId: futperform-club-manager

```



## Monitorização e Logs

### Firestore Console

#### 1. Functions

- Ver execuções e erros
- Métricas de performance
- Custos

#### 2. Firestore

- Uso de reads/writes
- Tamanho do database

#### 3. Authentication

- Utilizadores ativos
- Métodos de sign-in

## Logs

```
# Ver logs em tempo real
firebase functions:log

# Filtrar por function
firebase functions:log --only createDepartment

# Ver logs no console
# https://console.cloud.google.com/logs
```

## Segurança

### Checklist de Segurança

- [ ] Regras Firestore deployadas e testadas
- [ ] Custom claims configurados corretamente
- [ ] APIs keys restritas por domínio
- [ ] Storage rules configuradas
- [ ] CORS configurado nas functions
- [ ] Rate limiting ativado
- [ ] Audit logging ativo

### Restringir API Keys

No Firebase Console → Project Settings → API Keys:

- Restringir por domínios autorizados
- Ativar apenas serviços necessários

## Escalonamento

### Firestore

- Índices compostos para queries complexas
- Paginação em queries grandes
- Batch operations para writes múltiplos

### Functions

- Ajustar memory/timeout por function
- Implementar retry logic
- Usar Pub/Sub para tasks assíncronas

```
// Exemplo de configuração
exports.functionName = functions
  .runWith({
    memory: '512MB',
    timeoutSeconds: 60
  })
  .https.onCall(async (data, context) => {
    // ...
  });
```

## Gestão de Custos

---

### Firestore Pricing

- **Functions:** Gratuitas até 2M invocações/mês
- **Firestore:** Gratuito até 50K reads, 20K writes/dia
- **Hosting:** 10GB armazenamento, 360MB/dia gratuito
- **Auth:** Gratuito até 50K MAU (Google/Email)

### Otimizações

1. **Caching:** Usar Firestore offline persistence
2. **Batch Operations:** Agrupar reads/writes
3. **Índices:** Minimizar queries complexas
4. **CDN:** Usar Firebase Hosting CDN

## Troubleshooting

---

### Functions não deployam

```
# Verificar versão do Node
node --version # Deve ser 18+

# Reinstalar dependências
cd functions
rm -rf node_modules
npm install

# Verificar logs
firebase functions:log
```

### Regras Firestore rejeitam acesso

```
# Testar regras no simulador
# Firebase Console → Firestore → Rules → Playground

# Verificar custom claims do utilizador
# Firebase Console → Authentication → Users → Custom Claims
```

### Frontend não conecta ao Firebase

- Verificar credenciais em `lib/firebase.js`
- Verificar domínio autorizado em Firebase Console → Authentication → Settings
- Verificar CORS nas functions

## Recursos Adicionais

---

- [Firestore Documentation](https://firebase.google.com/docs) (<https://firebase.google.com/docs>)
  - [Next.js Deployment](https://nextjs.org/docs/deployment) (<https://nextjs.org/docs/deployment>)
  - [Firestore Best Practices](https://firebase.google.com/docs/firestore/best-practices) (<https://firebase.google.com/docs/firestore/best-practices>)
  - [Cloud Functions Guides](https://firebase.google.com/docs/functions) (<https://firebase.google.com/docs/functions>)
-



**Suporte:** Para questões técnicas, consultar a documentação ou contactar [suporte@futperformclub.pt](mailto:suporte@futperformclub.pt)