

# Funcionalidades - FutPerform Club Manager

---

Sistema completo de gestão de clubes de futebol com Firebase + Next.js

## Funcionalidades Implementadas

---

### Autenticação

- [x] Login com Email/Password
- [x] Login com Google Sign-In
- [x] Logout
- [x] Sessão persistente
- [x] Custom Claims (RBAC)
- role: array de papéis (Presidente, Treinador, Médico, etc.)
- departments: array de departamentos
- clubId: identificador do clube

### Gestão de Departamentos

- [x] **Listar Departamentos** (Organograma)
- Grid responsivo com cards
- Informações: nome, descrição, responsável, membros
- Contactos públicos (email, telefone)
- Lista de roles disponíveis
- Estatísticas (total membros, disponíveis, em campo)
- [x] **Criar Departamento** (Cloud Function)
- Nome, descrição, roles
- Contactos públicos
- Responsável (chefe do departamento)
- Feed inicial automático
- [x] **Pesquisa de Departamentos**
- Por nome
- Por descrição
- [x] **Departamentos Pré-configurados**
- Direção
- Equipa Técnica
- Clínico
- Nutrição
- Logística

## **Gestão de Membros**

- [x] **Listar Membros**
  - Por departamento
  - Informações: foto, nome, cargo, contacto
  - Estado de disponibilidade (badge colorido)
  - Permissões atribuídas
  - Último login
- [x] **Adicionar Membro** (Cloud Function)
  - Dados pessoais (nome, email, telefone)
  - Cargo/Role
  - Permissões específicas
  - Atribuição a departamento
  - Atualização automática de custom claims
- [x] **Atualizar Disponibilidade** (Cloud Function)
  - “Disponível” (verde)
  - “Em campo” (amarelo)
  - “De folga” (cinza)
- [x] **Perfil do Membro**
  - Foto de perfil
  - Informações de contacto
  - Lista de permissões
  - Histórico de atividades
- [x] **Filtros de Membros**
  - Ativos / Inativos
  - Por estado de disponibilidade
  - Por cargo/role

## **Gestão de Jogadores**

- [x] **Listar Jogadores**
  - Grid com cards responsivos
  - Foto, nome, posição, número
  - Estado clínico (Apto, Lesionado, Dúvida)
  - Idade calculada automaticamente
- [x] **Adicionar Jogador** (Cloud Function)
  - Nome completo
  - Data de nascimento
  - Posição
  - Número da camisola
  - Estado inicial

- [x] **Observações Multi-Departamento**
- Departamento médico: diagnósticos, tratamentos
- Departamento técnico: avaliações, desempenho
- Departamento nutricional: planos alimentares
- Autor e data de cada observação

- [x] **Adicionar Observação** (Cloud Function)

- Texto livre
- Departamento de origem
- Autor automático (do utilizador logado)
- Timestamp

- [x] **Filtros de Jogadores**

- Por posição (GR, Defesa, Médio, Avançado)
- Por estado clínico

- [x] **Estados Clínicos**

- Apto (verde)
- Lesionado (vermelho)
- Dúvida (amarelo)



## **Sistema de Requisições**

- [x] **Listar Requisições**
- Enviadas e recebidas
- Departamentos origem/destino
- Título e mensagem
- Estado (Pendente, Feito, Cancelado)
- Data de criação e atualização
- [x] **Enviar Requisição** (Cloud Function)
- Selecionar departamento destino
- Título e mensagem
- Notificação automática no feed do destino
- Registo completo com autor
- [x] **Atualizar Status** (Cloud Function)
- Marcar como “Feito”
- Marcar como “Cancelado”
- Timestamp de atualização
- [x] **Filtros de Requisições**
- Pendentes / Concluídas
- Por departamento
- Enviadas vs Recebidas

- [x] **Notificações**

- Post no feed do departamento de destino
- (TODO) Email para membros do departamento



## **Sistema de Feed**

- [x] **Feed Global**

- Todos os posts visíveis ao utilizador
- Baseado em departamentos, roles e users

- [x] **Feed por Departamento**

- Posts específicos do departamento
- Visibilidade controlada

- [x] **Criar Post** (Cloud Function)

- Título e corpo
- Departamento de origem
- Controlo de visibilidade granular:
  - departments: array de IDs
  - roles: array de papéis
  - users: array de UUIDs específicos

- [x] **Tipos de Posts**

- Comunicados gerais
- Atualizações médicas
- Avisos de treino
- Requisições interdepartamentais

- [x] **Metadados dos Posts**

- Autor (nome e UID)
- Departamento de origem
- Timestamp
- Visibilidade



## **Segurança (RBAC)**

- [x] **Firestore Security Rules**

- Acesso baseado em authentication
- Custom claims (role, departments)
- Direção/Presidente: acesso global
- Membros: acesso aos seus departamentos
- Feed: visibilidade controlada

- [x] **Níveis de Acesso**

1. Presidente/Direção: Tudo
2. Chefe de Departamento: Seu departamento completo
3. Membro: Permissões específicas

- [x] **Permissões Granulares**

- `read:players` - Ler dados de jogadores
- `write:players` - Editar jogadores
- `read:injuries` - Ler lesões
- `write:injuries` - Registrar lesões
- `read:training` - Ver planos de treino
- `write:training` - Criar/editar treinos
- `read:medical` - Aceder registos médicos
- `*` - Acesso total (apenas Direção)



## Interface do Utilizador

- [x] **Design Responsivo**

- Mobile-first
- Tablets
- Desktop

- [x] **Componentes UI**

- Cards interativos
- Badges de estado
- Modais para formulários
- Tooltips informativos
- Skeleton loaders
- Toasts de notificação

- [x] **Layout**

- Navbar com logo e perfil
- Sidebar navegável
- Breadcrumbs
- Footer

- [x] **Temas**

- Light mode (padrão)
- (TODO) Dark mode

- [x] **Acessibilidade**

- ARIA labels
- Keyboard navigation
- Focus indicators



## Cloud Functions

- [x] **createDepartment**

- Input: clubId, name, description, roles, contacts
- Output: departmentId
- Side-effects: Cria feed inicial

- [x] **addMemberToDepartment**
  - Input: clubId, departmentId, userId, role, permissions
  - Output: success
  - Side-effects: Atualiza custom claims, cria post no feed
- [x] **updateAvailability**
  - Input: clubId, departmentId, userId, status
  - Output: success
  - Side-effects: Atualiza timestamp
- [x] **sendDepartmentRequest**
  - Input: clubId, fromDept, toDept, message, title
  - Output: requestId
  - Side-effects: Cria post no feed de destino, (TODO) envia email
- [x] **updateRequestStatus**
  - Input: clubId, requestId, status
  - Output: success
- [x] **createFeedPost**
  - Input: clubId, departmentId, title, body, visibility
  - Output: postId
- [x] **addPlayer**
  - Input: clubId, name, dob, position, number, status
  - Output: playerId
- [x] **addPlayerObservation**
  - Input: clubId, playerId, departmentId, text
  - Output: success



## API Routes (Next.js)

- [x] `/api/departments` - GET, POST
- [x] `/api/members` - GET, POST
- [x] `/api/players` - GET, POST
- [x] `/api/requests` - GET, POST, PATCH
- [x] `/api/feed` - GET, POST



## Dados de Demonstração

- [x] **Seed Script**
  - 5 departamentos completos
  - 10+ membros com roles variados
  - 4 jogadores com observações
  - Posts de feed exemplo

- Requisições de teste



## Funcionalidades em Desenvolvimento

---

### Prioridade Alta

- ☐ **Notificações por Email**
  - Requisições entre departamentos
  - Novos posts no feed
  - Alterações em jogadores
- ☐ **Gestão de Treinos**
  - Planeamento de sessões
  - Participantes
  - Exercícios e durações
- ☐ **Calendário de Jogos**
  - Fixtures
  - Resultados
  - Convocatórias
- ☐ **Relatórios e Analytics**
  - Estatísticas de departamentos
  - Métricas de jogadores
  - Gráficos de lesões

### Prioridade Média

- ☐ **Chat Interno**
  - Mensagens diretas entre membros
  - Grupos por departamento
  - Notificações em tempo real
- ☐ **Upload de Documentos**
  - Firebase Storage
  - PDFs, imagens, vídeos
  - Organização por departamento
- ☐ **Gestão Financeira**
  - Orçamentos por departamento
  - Despesas e receitas
  - Relatórios financeiros
- ☐ **Módulo de Recrutamento**
  - Base de dados de potenciais jogadores
  - Avaliações

- Pipeline de contratação

## Prioridade Baixa

- [ ] **Dark Mode**
- [ ] **Multi-idioma** (EN, ES)
- [ ] **Exportação de Dados** (PDF, Excel)
- [ ] **Integração com APIs Externas**
- Transfermarkt
- SofaScore
- [ ] **App Mobile** (React Native / Flutter)



## Bugs Conhecidos

- Nenhum bug crítico reportado



## Melhorias Sugeridas

### 1. Performance

- Implementar paginação em listas grandes
- Cache de dados frequentes
- Lazy loading de imagens

### 2. UX

- Adicionar tours interativos para novos utilizadores
- Keyboard shortcuts
- Undo/Redo em formulários

### 3. Segurança

- Two-factor authentication
- Audit logs detalhados
- Rate limiting nas API calls

### 4. Analytics

- Google Analytics integrado
- Eventos customizados
- Heatmaps



## Notas Técnicas

### Stack Tecnológica

- **Frontend:** Next.js 14, React 18, TypeScript
- **Backend:** Firebase (Auth, Firestore, Functions, Storage, Hosting)
- **Styling:** Tailwind CSS, shadcn/ui
- **Deployment:** Firebase Hosting, Vercel (opção)



## Estrutura de Dados

```
clubs/{clubId}
├── departments/{departmentId}
│   ├── name, description, head, roles
│   ├── contacts_public: {email, phone}
│   └── members/{memberId}
│       ├── uid, name, role, permissions
│       ├── active, availableStatus
│       └── lastLogin, joinedAt
├── players/{playerId}
│   ├── name, dob, position, number, status
│   └── observations: [{dept, text, date, author}]
├── feed/{postId}
│   ├── author, title, body, departmentId
│   └── visibility: {departments, roles, users}
└── requests/{requestId}
    ├── fromDept, toDept, message, status
    └── author, createdAt, updatedAt
```

## Convenções

- **Idioma:** Português de Portugal
- **Date Format:** DD/MM/YYYY
- **Phone Format:** +351 9XX XXX XXX
- **Email:** nome.sobrenome@clube.pt

---

**Última Atualização:** Janeiro 2025

**Versão:** 1.0.0