

ESPECIFICAÇÕES TÉCNICAS - SISTEMA YUNA

ARQUITETURA DE SOFTWARE

Padrão Arquitetural

Multi-SPA (Multiple Single Page Applications) - Cada diretório funciona como uma SPA independente - Compartilhamento de recursos via arquivos na raiz - Deploy otimizado com cache granular - Manutenção modular facilitada

Stack Tecnológico Detalhado

Frontend:

```
// Tecnologias Core
HTML5: Estrutura semântica
CSS3: Estilização responsiva + Grid/Flexbox
JavaScript ES6+: Lógica de negócio vanilla
```

// Bibliotecas Externas

```
Firebase SDK 9.23.0: Backend-as-a-Service
Font Awesome 6.4.0: Iconografia
XLSX.js: Exportação de dados
EmailJS: Sistema de notificações
```

Backend (Serverless):

```
// Firebase Services
Authentication: Gerenciamento de usuários
Firestore: Banco NoSQL em tempo real
Hosting: Deploy e CDN global
Functions: Processamento serverless (futuro)
```

// Configuração de Segurança

```
Rules: Validação server-side
Indexes: Otimização de queries
```

Estrutura de Dados (Firestore)

Coleções Principais:

```
// usuarios_admin
{
  uid: string,
  email: string,
  nome: string,
  role: 'super_admin' | 'admin',
}
```

```

    ativo: boolean,
    criadoEm: timestamp,
    criadoPor: string,
    permissoes: {
        criarUsuarios: boolean,
        verRelatorios: boolean,
        verPesquisaSatisfacao: boolean
    }
}

// usuarios_equipe
{
    uid: string,
    email: string,
    nome: string,
    equipe: 'manutencao' | 'nutricao' | 'higienizacao' | 'hotelaria',
    role: 'equipe',
    ativo: boolean,
    criadoEm: timestamp
}

// usuarios_acompanhantes
{
    uid: string,
    email: string,
    nome: string,
    quarto: string,
    telefone: string,
    role: 'acompanhante',
    ativo: boolean,
    criadoEm: timestamp
}

// solicitacoes
{
    id: string,
    tipo: string,
    equipe: string,
    status: 'pendente' | 'em-andamento' | 'finalizada',
    prioridade: 'normal' | 'alta' | 'urgente',
    quarto: string,
    descricao: string,
    usuarioNome: string,
    usuarioUid: string,
    dataAbertura: timestamp,
    dataFinalizacao?: timestamp,

```

```

    responsavel?: string,
    solucao?: string,
    tempoAtendimentoMinutos?: number
  }

```

```

// avaliacoes_satisfacao
{
  id: string,
  solicitacaoId: string,
  nota: number (1-5),
  comentario?: string,
  dataAvaliacao: timestamp,
  usuarioUid: string
}

```

SISTEMA DE SEGURANÇA

Autenticação

```

// Firebase Authentication
- OAuth2 + JWT tokens
- Session management automático
- Password policies configuráveis
- Rate limiting integrado

```

Autorização (RBAC)

```

// Role-Based Access Control
function verificarPermissoes(usuario, acao) {
  const permissoes = {
    super_admin: ['*'], // Todas as permissões
    admin: ['view_all', 'export_reports', 'view_satisfaction'],
    equipe: ['view_own_team', 'update_requests'],
    acompanhante: ['create_request', 'view_own_requests']
  };

  return permissoes[usuario.role].includes(acao) ||
    permissoes[usuario.role].includes('*');
}

```

Firestore Security Rules

```

// Regras de segurança no servidor
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // Solicitações - filtro por equipe

```

```

    match /solicitacoes/{document} {
      allow read: if isAuthenticated() &&
        (isSuperAdmin() || isAdmin() ||
          (isEquipe() && resource.data.equipe == getUser().equipe) ||
          (isAcompanhante() && resource.data.usuarioUid == request.auth.uid));

      allow write: if isAuthenticated() &&
        (isSuperAdmin() ||
          (isEquipe() && resource.data.equipe == getUser().equipe) ||
          (isAcompanhante() && request.auth.uid != null));
    }
  }
}

```

PERFORMANCE E OTIMIZAÇÃO

PWA (Progressive Web App)

```

// Service Worker Strategy
const CACHE_NAME = 'yuna-v1.0';
const urlsToCache = [
  '/admin/',
  '/acompanhantes/',
  '/firebase-config-secure.js',
  '/manifest.json'
];

// Cache Strategy: Network First para dados dinâmicos
// Cache Strategy: Cache First para assets estáticos

```

Otimizações Implementadas:

- **Lazy Loading:** Carregamento sob demanda
- **Code Splitting:** Divisão de código por módulos
- **Minificação:** Compressão CSS/JS automática
- **CDN:** Distribuição global via Netlify
- **Gzip:** Compressão de transferência
- **Critical CSS:** CSS inline para above-the-fold

Métricas de Performance:

```

// Core Web Vitals Targets
LCP (Largest Contentful Paint): < 2.5s
FID (First Input Delay): < 100ms
CLS (Cumulative Layout Shift): < 0.1

```

```
// Additional Metrics
TTFB (Time to First Byte): < 200ms
Speed Index: < 3.0s
Total Blocking Time: < 200ms
```

RESPONSIVIDADE E UX

Breakpoints CSS:

```
/* Mobile First Approach */
@media (min-width: 768px) { /* Tablet */ }
@media (min-width: 1024px) { /* Desktop */ }
@media (min-width: 1440px) { /* Large Desktop */ }

/* High DPI Displays */
@media (-webkit-min-device-pixel-ratio: 2) { /* Retina */ }
```

Design System:

```
:root {
  /* Cores Primárias */
  --primary-blue: #3b82f6;
  --primary-green: #10b981;
  --primary-orange: #f59e0b;

  /* Cores por Equipe */
  --manutencao: #ef4444;
  --nutricao: #22c55e;
  --higienizacao: #3b82f6;
  --hotelaria: #a855f7;

  /* Tipografia */
  --font-family: 'Inter', -apple-system, BlinkMacSystemFont, sans-serif;
  --font-size-base: 16px;
  --line-height: 1.5;

  /* Spacing System */
  --spacing-xs: 0.25rem;
  --spacing-sm: 0.5rem;
  --spacing-md: 1rem;
  --spacing-lg: 1.5rem;
  --spacing-xl: 2rem;
}
```

INTEGRAÇÃO E APIs

Firestore Integration:

```
// Configuração Firebase
const firebaseConfig = {
  apiKey: process.env.FIREBASE_API_KEY,
  authDomain: "studio-5526632052-23813.firebaseio.com",
  projectId: "studio-5526632052-23813",
  storageBucket: "studio-5526632052-23813.appspot.com",
  messagingSenderId: process.env.FIREBASE_MESSAGING_SENDER_ID,
  appId: process.env.FIREBASE_APP_ID
};

// Real-time Listeners
const unsubscribe = db.collection('solicitacoes')
  .where('equipe', '==', userEquipe)
  .onSnapshot((snapshot) => {
    snapshot.docChanges().forEach((change) => {
      if (change.type === 'added') {
        handleNewRequest(change.doc.data());
      }
      if (change.type === 'modified') {
        handleRequestUpdate(change.doc.data());
      }
    });
  });
```

EmailJS Integration:

```
// Sistema de Notificações
const emailParams = {
  to_email: userEmail,
  from_name: 'Sistema YUNA',
  subject: 'Nova Solicitação de Serviço',
  message: `Solicitação #${requestId} foi ${status}`
};

emailjs.send(SERVICE_ID, TEMPLATE_ID, emailParams, PUBLIC_KEY);
```

DEPLOYMENT E CI/CD

Environment Configuration:

```
// Netlify Deploy Settings
[build]
  publish = "."
```

```

    command = "echo 'Static site - no build required'"

[[redirects]]
  from = "/admin/*"
  to = "/admin/index.html"
  status = 200

[[redirects]]
  from = "/acompanhantes/*"
  to = "/acompanhantes/index.html"
  status = 200

// API Proxy
[[redirects]]
  from = "/api/*"
  to = "https://api.yuna.com.br/:splat"
  status = 200

```

GitHub Actions Pipeline:

```

name: Deploy to Production
on:
  push:
    branches: [ main ]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Deploy to Netlify
        uses: netlify/actions/build@master
    env:
      NETLIFY_AUTH_TOKEN: ${ secrets.NETLIFY_AUTH_TOKEN }
      NETLIFY_SITE_ID: ${ secrets.NETLIFY_SITE_ID }

```

MONITORING E ANALYTICS

Error Tracking:

```

// Custom Error Handler
window.addEventListener('error', (event) => {
  console.error('[SISTEMA-ERRO]', {
    message: event.message,
    filename: event.filename,
    lineno: event.lineno,
    colno: event.colno,

```

```

        error: event.error,
        timestamp: new Date().toISOString(),
        userAgent: navigator.userAgent,
        url: window.location.href
    });

    // Enviar para sistema de monitoramento
    sendErrorToMonitoring(event);
});

```

Performance Monitoring:

```

// Web Vitals Tracking
import { getCLS, getFID, getFCP, getLCP, getTTFB } from 'web-vitals';

getCLS(sendToAnalytics);
getFID(sendToAnalytics);
getFCP(sendToAnalytics);
getLCP(sendToAnalytics);
getTTFB(sendToAnalytics);

```

ESCALABILIDADE E FUTURO

Arquitetura Preparada Para:

- **Microservices:** Migração gradual para arquitetura distribuída
- **Multi-tenant:** Suporte a múltiplas clínicas/hospitais
- **Mobile Apps:** Base PWA facilita desenvolvimento nativo
- **AI/ML:** APIs preparadas para integração de IA
- **IoT:** Integração com dispositivos hospitalares

Roadmap Técnico:

1. **v1.1:** Mobile apps nativas (React Native)
2. **v1.2:** Sistema de relatórios avançado (BI)
3. **v2.0:** Multi-tenant completo
4. **v2.1:** Integração HL7/FHIR
5. **v3.0:** AI para predição de demandas

© 2024-2025 Samuel dos Reis Lacerda Junior - Todos os direitos reservados

Sistema YUNA - Especificações Técnicas Proprietárias

Desenvolvido por: Samuel dos Reis Lacerda Junior

CNPJ: 55.004.442 SAMUEL DOS REIS LACERDA JUNIOR

Contato: samukajr82@gmail.com | +55 11 94586-4671