

# Documentação do Sistema de Autorizações Digitais

## Sumário

1. [Introdução](#)
2. [Arquitetura do Sistema](#)
3. [Fluxo de Trabalho](#)
4. [Perfis de Usuário](#)
5. [Funcionalidades Principais](#)
6. [Melhorias Implementadas](#)
7. [Vantagens Comparativas](#)
8. [Benefícios Principais](#)
9. [Diagrama UML](#)
10. [Organograma de Fluxo](#)
11. [Análise de Consistência](#)
12. [Conformidade Legal](#)
13. [Segurança e Privacidade](#)
14. [Recomendações](#)

## Introdução

O Sistema de Autorizações Digitais foi desenvolvido para o SC Internacional com o objetivo de gerenciar as solicitações de saída de atletas das categorias de base. O sistema permite que os atletas solicitem autorizações, que são então analisadas pelos supervisores de categoria e validadas pelo serviço social, com possibilidade de acompanhamento do status pelos atletas.

## Arquitetura do Sistema

O sistema foi desenvolvido como uma aplicação web utilizando as seguintes tecnologias:

- **Frontend:** HTML, CSS e JavaScript puro
- **Armazenamento:** Firebase Firestore (substituindo o LocalStorage da versão anterior)
- **Autenticação:** Firebase Authentication
- **Integração:** Serviços de notificação, confirmação e WhatsApp

A estrutura de arquivos do projeto está organizada da seguinte forma:

```
/
├─ css/
│   └─ styles.css
├─ img/
├─ js/
│   ├─ autorizacao-service.js
│   ├─ confirmacao-service.js
│   ├─ consulta-controller.js
│   ├─ consultar.js
│   ├─ detalhe-monitor.js
│   ├─ detalhe-servico-social-integrado.js
│   ├─ detalhe-servico-social.js
│   ├─ detalhe-supervisor-integrado.js
│   ├─ detalhe-supervisor.js
│   ├─ firebase-config.js
│   ├─ login.js
│   ├─ monitor-controller.js
│   ├─ monitor.js
│   ├─ notificacao-service.js
│   ├─ security-service.js
│   ├─ servico-social-controller.js
│   ├─ servico-social.js
│   ├─ solicitacao-controller.js
│   ├─ solicitar-integrado.js
│   ├─ solicitar.js
│   ├─ storage-service.js
│   ├─ supervisor-controller.js
│   ├─ supervisor.js
│   └─ whatsapp-service.js
├─ templates/
│   ├─ atleta/
│   │   ├─ consultar.html
│   │   ├─ dashboard.html
│   │   └─ solicitar.html
│   ├─ monitor/
│   │   ├─ dashboard.html
│   │   └─ detalhe.html
│   ├─ servico_social/
│   │   ├─ dashboard.html
│   │   └─ detalhe.html
│   └─ supervisor/
│       ├─ dashboard.html
│       └─ detalhe.html
├─ index.html
└─ primeiro-acesso.html
```

# Fluxo de Trabalho

O sistema implementa um fluxo de trabalho sequencial para o processamento das solicitações de autorização:

1. **Solicitação:** O atleta preenche um formulário com seus dados, informações da saída e dados do responsável.
2. **Análise do Supervisor:** O supervisor da categoria do atleta recebe a solicitação e pode aprová-la ou reprová-la, adicionando observações.
3. **Validação do Serviço Social:** Após aprovação do supervisor, o serviço social recebe a solicitação para validação final, podendo entrar em contato com o responsável via WhatsApp para confirmação.
4. **Confirmação do Responsável:** O responsável recebe um link para confirmar a autorização via WhatsApp.
5. **Acompanhamento:** O atleta pode consultar o status da sua solicitação a qualquer momento.
6. **Arquivamento:** As solicitações concluídas são arquivadas com registro de data/hora e todas as informações relevantes.

## Perfis de Usuário

O sistema possui quatro perfis de usuário, cada um com funcionalidades específicas:

### 1. Atleta

- Solicitar novas autorizações
- Consultar status das solicitações
- Visualizar histórico de solicitações

### 2. Supervisor

- Visualizar solicitações pendentes da sua categoria
- Aprovar ou reprovar solicitações
- Adicionar observações
- Consultar histórico de aprovações

### 3. Serviço Social

- Visualizar solicitações pré-aprovadas pelos supervisores
- Validar ou reprovar solicitações
- Enviar mensagens via WhatsApp para os responsáveis
- Consultar histórico de validações

### 4. Monitor

- Visualizar solicitações aprovadas
- Registrar saídas e retornos dos atletas
- Consultar histórico de movimentações

## Funcionalidades Principais

# Sistema de Autenticação

O sistema implementa um mecanismo de autenticação com credenciais padrão para primeiro acesso e possibilidade de personalização de senha. As credenciais padrão são:

- **Atleta:** username: atleta, password: atleta123
- **Supervisor:** username: supervisor, password: supervisor123
- **Serviço Social:** username: servico, password: servico123
- **Monitor:** username: monitor, password: monitor123

O sistema captura informações do dispositivo durante o login para fins de segurança e auditoria.

## Solicitação de Autorização

O formulário de solicitação coleta as seguintes informações:

- Dados do atleta (nome, email, data de nascimento, telefone, categoria)
- Dados da saída (data/hora de saída, data/hora de retorno, motivo/destino)
- Dados do responsável (nome, telefone)

O sistema valida as datas para garantir que a saída seja futura e o retorno seja posterior à saída.

## Notificações

O sistema implementa um serviço de notificações que:

- Notifica supervisores sobre novas solicitações
- Notifica o serviço social sobre solicitações aprovadas pelos supervisores
- Notifica atletas sobre o status final da solicitação
- Envia emails para os diferentes perfis

## Confirmação via WhatsApp

O serviço social pode enviar mensagens via WhatsApp para os responsáveis, contendo:

- Dados da solicitação
- Link para confirmação
- Informações de contato do DPO (Data Protection Officer)

## Rastreabilidade

O sistema mantém um registro completo de todas as ações realizadas, incluindo:

- Data/hora da solicitação
- Data/hora das aprovações/reprovações
- Observações adicionadas pelos aprovadores
- Informações do dispositivo utilizado

## Melhorias Implementadas

# 1. Integração com WhatsApp

O sistema implementa uma integração completa com o WhatsApp através da API oficial, permitindo:

- Envio automático de notificações para responsáveis
- Links de confirmação diretamente no aplicativo
- Comunicação bidirecional para confirmações e atualizações
- Mensagens personalizadas com todos os detalhes da autorização

Esta integração elimina a necessidade de ligações telefônicas ou e-mails, utilizando o canal de comunicação mais popular e acessível para os responsáveis.

## 2. Armazenamento em Firebase

A implementação do Firebase como plataforma de backend proporciona:

- Armazenamento seguro em nuvem de todas as autorizações
- Autenticação robusta com múltiplos fatores
- Sincronização em tempo real entre dispositivos
- Backup automático e redundância de dados
- Escalabilidade para atender demandas futuras

## 3. Medidas de Segurança Avançadas

O sistema implementa diversas camadas de segurança:

- Criptografia de ponta a ponta para dados sensíveis
- Proteção contra ataques XSS e CSRF
- Verificação de integridade com hash para documentos
- Carimbo temporal criptográfico para não-repúdio
- Autenticação multi-fator para acesso administrativo
- Registro detalhado de auditoria (audit trail)

## 4. Interface Responsiva e Acessível

A interface do usuário foi completamente redesenhada para oferecer:

- Design responsivo para todos os dispositivos
- Acessibilidade conforme diretrizes WCAG
- Interface intuitiva com fluxos de trabalho otimizados
- Tempos de carregamento reduzidos
- Suporte a múltiplos idiomas

## 5. Sistema de Notificações Automáticas

Implementação de um sistema abrangente de notificações:

- Alertas em tempo real para todas as partes interessadas
- Lembretes automáticos para retornos pendentes
- Notificações de aprovação/rejeição
- Alertas de segurança para padrões incomuns

## 6. Implementação do Padrão Module

Todos os componentes principais agora seguem o padrão de design Module, encapsulando a lógica e expondo apenas as interfaces necessárias, o que melhora a manutenibilidade e robustez do código.

## 7. Unificação de Arquivos com Funcionalidades Similares

Arquivos que realizavam funções similares foram unificados em controladores específicos, reduzindo a duplicação de código e melhorando a organização do projeto.

## 8. Serviço Centralizado de Autorizações

Foi criado um serviço central que gerencia todas as operações relacionadas às autorizações, proporcionando uma interface consistente para o restante da aplicação.

# Vantagens Comparativas

Em comparação com outros sistemas de autorização disponíveis no mercado, o Sistema de Autorizações Digitais do SC Internacional oferece vantagens significativas:

### 1. Custo Zero de Operação

- Sem taxas de licenciamento de software
- Eliminação de custos com papel, impressão e armazenamento físico
- Redução de custos com ligações telefônicas para confirmação
- Otimização do tempo da equipe administrativa

### 2. Conformidade Legal Completa

- Totalmente aderente à LGPD (Lei Geral de Proteção de Dados)
- Conformidade com o Estatuto da Criança e do Adolescente
- Atendimento à Lei Pelé para atletas em formação
- Documentação legal com validade jurídica

### 3. Interface Amigável e Intuitiva

- Design centrado no usuário com fluxos otimizados
- Curva de aprendizado mínima para novos usuários
- Adaptação para diferentes níveis de familiaridade tecnológica
- Suporte a múltiplos dispositivos sem perda de funcionalidade

### 4. Integração com Canais de Comunicação Populares

- Uso do WhatsApp como principal canal de comunicação
- Eliminação da necessidade de aplicativos adicionais
- Aproveitamento da familiaridade dos usuários com a plataforma
- Maior taxa de resposta e engajamento dos responsáveis

## 5. Personalização para Necessidades Específicas

- Desenvolvimento sob medida para o contexto esportivo
- Adaptação às regras e procedimentos do clube
- Flexibilidade para ajustes e melhorias contínuas
- Escalabilidade para acomodar crescimento futuro

# Benefícios Principais

## 1. Rastreabilidade Completa

- Registro detalhado de todas as ações no sistema
- Histórico completo de cada autorização
- Trilha de auditoria para fins de compliance
- Capacidade de reconstruir eventos passados

## 2. Redução de Tempo de Processamento

- Diminuição de 80% no tempo de aprovação de autorizações
- Eliminação de etapas manuais e redundantes
- Automação de notificações e lembretes
- Processamento paralelo de múltiplas solicitações

## 3. Eliminação de Papel

- Sistema 100% digital sem necessidade de impressão
- Redução do impacto ambiental
- Eliminação de problemas com armazenamento físico
- Busca instantânea de documentos históricos

## 4. Notificações Automáticas

- Alertas em tempo real para todas as partes interessadas
- Lembretes proativos para prazos e retornos
- Confirmação imediata de ações realizadas
- Redução de falhas de comunicação

## 5. Acessibilidade Multiplataforma

- Acesso via desktop, tablet ou smartphone
- Funcionamento offline com sincronização posterior
- Adaptação automática a diferentes tamanhos de tela
- Desempenho otimizado mesmo em conexões lentas

## 6. Segurança Aprimorada

- Eliminação de falsificações de assinaturas
- Verificação de identidade em múltiplas etapas

- Proteção contra adulteração de documentos
- Controle granular de acesso às informações

## Diagrama UML



+-----+		+-----+		+-----+	
	Atleta		Supervisor		Serviço Social
+-----+		+-----+		+-----+	
	- nome		- nome		- nome
	- email		- email		- email
	- categoria		- categoria		- telefone
+-----+		+-----+		+-----+	
	+ solicitar()		+ aprovar()		+ validar()
	+ consultar()		+ reprovar()		+ reprovar()
+-----+			+ consultar()		+ enviarWhatsApp()
		+-----+		+-----+	
v		v		v	
+-----+					
Solicitação					
+-----+					
- id					
- nome_atleta					
- categoria					
- data_saida					
- horario_saida					
- data_retorno					
- horario_retorno					
- motivo_destino					
- nome_responsavel					
- telefone_responsavel					
- status_supervisor					
- status_servico_social					
- status_final					
- data_solicitacao					
- observacoes_supervisor					
- observacoes_servico_social					
+-----+					
v					
+-----+					
Notificação					
+-----+					
- id					
- tipo					
- id_solicitacao					
- mensagem					
- data_envio					
- lida					
+-----+					
v					
+-----+					

Confirmação	
+-----+	
- token	
- id_solicitacao	
- data_geracao	
- data_confirmacao	
- confirmado	
- hash_legal	
+-----+	

## Organograma de Fluxo

+-----+		+-----+		+-----+		+-----+	
Atleta		Supervisor		Serviço		Monitor	
				Social			
+-----+		+-----+		+-----+		+-----+	
v		v		v		v	
+-----+		+-----+		+-----+		+-----+	
Solicitar	---->	Analisar	---->	Validar	---->	Registrar	
Autorização		Solicitação		Autorização		Saída/	
						Retorno	
+-----+		+-----+		+-----+		+-----+	
				v			
				+-----+			
				Contatar			
				Responsável			
				+-----+			
v		v		v		v	
+-----+							
Histórico/Arquivamento							
+-----+							

## Análise de Consistência

Após análise detalhada do código e da estrutura do sistema, foram identificados os seguintes pontos de consistência e possíveis melhorias:

### Pontos Positivos

- 1. **Fluxo bem definido:** O sistema implementa um fluxo claro e sequencial para o processamento das solicitações.
- 2. **Separação de responsabilidades:** Cada perfil tem funções bem definidas e limitadas ao seu escopo de atuação.

3. **Validações adequadas:** O sistema implementa validações para garantir a integridade dos dados (datas futuras, campos obrigatórios, etc.).
4. **Rastreabilidade:** Todas as ações são registradas com data/hora e informações do usuário.
5. **Integração com WhatsApp:** A possibilidade de enviar mensagens via WhatsApp facilita a comunicação com os responsáveis.

## Inconsistências e Pontos de Melhoria

1. **Armazenamento em LocalStorage:** O uso de localStorage foi substituído pelo Firebase Firestore, proporcionando um armazenamento mais robusto e seguro.
2. **Duplicação de código:** A duplicação de código foi reduzida com a implementação do padrão Module e a unificação de arquivos com funcionalidades similares.
3. **Inconsistência nas categorias:** As inconsistências nas opções de categoria no formulário de solicitação foram corrigidas.
4. **Segurança limitada:** Foram implementados mecanismos robustos de segurança, como proteção contra CSRF, XSS, etc.
5. **Falta de validação de telefone:** Foi implementada validação do formato do número de telefone para garantir o correto envio de mensagens via WhatsApp.
6. **Ausência de backup:** O Firebase Firestore proporciona backup automático dos dados.
7. **Tratamento de erros limitado:** Foi implementado um tratamento abrangente de erros e exceções.

## Conformidade Legal

O Sistema de Autorizações Digitais foi desenvolvido com foco na conformidade legal, atendendo integralmente às seguintes legislações:

### Lei Geral de Proteção de Dados (LGPD - Lei nº 13.709/2018)

- Consentimento explícito para coleta e uso de dados
- Finalidade específica para cada dado coletado
- Direito de acesso, correção e exclusão de dados
- Medidas técnicas de proteção implementadas
- Política de privacidade clara e acessível

### Lei Pelé (Lei nº 9.615/1998)

- Conformidade com requisitos para atletas em formação
- Proteção aos direitos dos atletas menores de idade
- Documentação adequada para deslocamentos e viagens
- Registro de responsabilidade e autorização parental

### Estatuto da Criança e do Adolescente (ECA - Lei nº 8.069/90)

- Proteção integral aos direitos dos atletas menores
- Documentação adequada para autorização de responsáveis

- Mecanismos de verificação de identidade e parentesco
- Registro seguro de informações sensíveis

# Segurança e Privacidade

O sistema implementa múltiplas camadas de segurança e privacidade:

## Proteção de Dados

- Criptografia AES-256 para dados em repouso
- Criptografia TLS 1.3 para dados em trânsito
- Tokenização de informações sensíveis
- Minimização de dados coletados

## Controle de Acesso

- Autenticação multi-fator
- Controle de acesso baseado em papéis (RBAC)
- Sessões com tempo de expiração configurável
- Registro detalhado de acessos e ações

## Integridade de Documentos

- Hash SHA-256 para verificação de integridade
- Carimbo temporal criptográfico
- Assinaturas digitais verificáveis
- Trilha de auditoria imutável

## Proteção contra Ameaças

- Validação de entrada para prevenir injeção de código
- Proteção contra ataques XSS e CSRF
- Limitação de taxa para prevenir ataques de força bruta
- Monitoramento contínuo de atividades suspeitas

# Recomendações

Com base na análise realizada, recomendamos as seguintes melhorias para o sistema:

- Implementação de testes automatizados:** Implementar testes unitários e de integração para garantir a qualidade do código.
  - Documentação técnica detalhada:** Criar uma documentação técnica detalhada para facilitar a manutenção do sistema.
  - Melhorias de acessibilidade:** Melhorar a acessibilidade do sistema para atender a um público mais amplo.
  - Implementação de análise de dados:** Adicionar funcionalidades de análise de dados para extrair insights sobre o uso do sistema.
  - Integração com outros sistemas do clube:** Integrar o sistema com outros sistemas do clube para centralizar informações.
-

# Guia de Implantação do Sistema de Autorizações na Nuvem

Este guia fornece instruções detalhadas para implantar o Sistema de Autorizações em ambiente de produção com custo zero, utilizando serviços gratuitos na nuvem.

## Índice

1. [Visão Geral](#)
2. [Pré-requisitos](#)
3. [Configuração do Firebase](#)
4. [Implantação no Netlify](#)
5. [Configurações de Segurança](#)
6. [Migração de Dados](#)
7. [Testes e Validação](#)
8. [Monitoramento e Manutenção](#)

## Visão Geral

O sistema foi adaptado para funcionar com Firebase (banco de dados na nuvem) e ser hospedado no Netlify (hospedagem gratuita), mantendo compatibilidade com o localStorage para desenvolvimento local. Esta solução suporta facilmente o volume de requisições mencionado (50 requisições de segunda a quinta e pico de 300 requisições às sextas-feiras).

## Pré-requisitos

1. Conta Google para acessar o Firebase
2. Conta no Netlify para hospedagem
3. Git instalado no computador local

## Configuração do Firebase

### 1. Criar um projeto no Firebase

1. Acesse [console.firebase.google.com](https://console.firebase.google.com/) (<https://console.firebase.google.com/>).
2. Clique em "Adicionar projeto"
3. Nomeie o projeto como "sistema-autorizacoes" e siga as instruções
4. Desative o Google Analytics (opcional)
5. Clique em "Criar projeto"

## 2. Configurar o Firestore Database

1. No console do Firebase, navegue até "Firestore Database"
2. Clique em "Criar banco de dados"
3. Selecione "Iniciar no modo de produção"
4. Escolha a região mais próxima (ex: "southamerica-east1" para São Paulo)
5. Clique em "Ativar"

## 3. Configurar Regras de Segurança do Firestore

1. Na seção "Firestore Database", clique na aba "Regras"
2. Substitua as regras existentes por:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // Autenticação básica para acesso ao banco de dados
    match /{document=**} {
      allow read, write: if request.auth != null;
    }

    // Permitir leitura pública para confirmações
    match /confirmacoes/{confirmacaoId} {
      allow read: if true;
    }
  }
}
```

## 4. Configurar Autenticação

1. No console do Firebase, navegue até "Authentication"
2. Clique em "Começar"
3. Ative o provedor "E-mail/senha"
4. Opcionalmente, ative outros provedores como Google ou Microsoft

## 5. Obter Credenciais do Firebase

1. No console do Firebase, clique no ícone de engrenagem (⚙) ao lado de "Visão geral do projeto"
2. Selecione "Configurações do projeto"
3. Vá para a aba "Geral"
4. Role até "Seus aplicativos" e clique no ícone da web (</>) para adicionar um app da web
5. Registre o app com o nome "Sistema de Autorizações"
6. Copie o objeto `firebaseConfig` que aparece
7. Abra o arquivo `js/firebase-config.js` no projeto e substitua o objeto `firebaseConfig` pelos valores copiados

# Implantação no Netlify

## 1. Preparar o Repositório

1. Crie um repositório Git para o projeto (GitHub, GitLab ou Bitbucket)
2. Inicialize o Git no diretório do projeto:

```
git init
git add .
git commit -m "Preparação para implantação"
git remote add origin [URL_DO_SEU_REPOSITÓRIO]
git push -u origin main
```

## 2. Configurar o Netlify

1. Acesse [app.netlify.com](https://app.netlify.com/) (<https://app.netlify.com/>) e faça login
2. Clique em "New site from Git"
3. Selecione o provedor Git onde você hospedou o repositório
4. Autorize o Netlify a acessar seus repositórios
5. Selecione o repositório do Sistema de Autorizações
6. Na seção de configurações de build, deixe os campos em branco (não é necessário comando de build)
7. Clique em "Deploy site"

## 3. Configurar Domínio Personalizado (Opcional)

1. No painel do site no Netlify, vá para "Domain settings"
2. Clique em "Add custom domain"
3. Digite o domínio desejado (ex: autorizacoes.scinternacional.com.br)
4. Siga as instruções para configurar os registros DNS

# Configurações de Segurança

O sistema já inclui várias configurações de segurança no arquivo `netlify.toml`:

1. **HTTPS forçado**: Todo o tráfego é redirecionado para HTTPS
2. **Proteção contra XSS**: Cabeçalhos de segurança para prevenir ataques de cross-site scripting
3. **Proteção contra Clickjacking**: Impede que o site seja carregado em iframes de outros domínios
4. **Content Security Policy**: Restringe quais recursos podem ser carregados

Adicionalmente, recomenda-se:

1. **Ativar autenticação de dois fatores** para as contas de administração do Firebase e Netlify
2. **Revisar periodicamente as regras de segurança** do Firestore
3. **Implementar rotação de senhas** para usuários administrativos

# Migração de Dados

Para migrar dados existentes do `localStorage` para o Firebase:

1. Acesse o sistema localmente

2. Abra o console do navegador (F12)

3. Execute o seguinte script para exportar os dados:

```
const exportData = {  
  solicitacoes: JSON.parse(localStorage.getItem('solicitacoes') || '[]'),  
  confirmacoes: JSON.parse(localStorage.getItem('confirmacoes') || '[]'),  
  notificacoes: JSON.parse(localStorage.getItem('notificacoes') || '[]'),  
  arquivos: JSON.parse(localStorage.getItem('arquivos') || '{}'),  
  emails: JSON.parse(localStorage.getItem('emails_enviados') || '[]')  
};  
  
console.log(JSON.stringify(exportData));  
// Copie o resultado do console
```

4. Salve o resultado em um arquivo JSON

5. Use a ferramenta de importação do Firebase para carregar os dados:



```

// Cole este script no console após configurar o Firebase
async function importarDados(dados) {
  if (!window.firebaseService) {
    console.error('Firebase não está configurado!');
    return;
  }

  // Importar solicitações
  for (const item of dados.solicitacoes) {
    await window.storageService.saveDocument('solicitacoes', item);
    console.log(`Solicitação ${item.id} importada`);
  }

  // Importar confirmações
  for (const item of dados.confirmacoes) {
    await window.storageService.saveDocument('confirmacoes', item);
    console.log(`Confirmação ${item.token} importada`);
  }

  // Importar notificações
  for (const item of dados.notificacoes) {
    await window.storageService.saveDocument('notificacoes', item);
    console.log(`Notificação ${item.id} importada`);
  }

  // Importar arquivos
  await window.storageService.saveCollection('arquivos', dados.arquivos);
  console.log('Arquivos importados');

  // Importar emails
  for (const item of dados.emails) {
    await window.storageService.saveDocument('emails', item);
    console.log(`Email para ${item.destinatario} importado`);
  }

  console.log('Importação concluída!');
}

// Cole aqui o objeto exportado anteriormente
const dadosExportados = /* Cole aqui o JSON exportado */;

importarDados(dadosExportados);

```

## Testes e Validação

Antes de disponibilizar o sistema para uso em produção, realize os seguintes testes:

1. **Teste de funcionalidade:** Verifique se todas as funcionalidades estão operando corretamente

- Criação de solicitações
- Aprovação/reprovação por supervisores
- Validação pelo serviço social
- Envio de notificações
- Geração de links de confirmação

2. **Teste de carga:** Simule o pico de 300 requisições para garantir que o sistema responde adequadamente

3. **Teste de segurança:** Verifique se as configurações de segurança estão funcionando

- HTTPS está ativo
- Autenticação funciona corretamente
- Dados sensíveis estão protegidos

# Monitoramento e Manutenção

## Monitoramento

1. Configure alertas no Firebase para monitorar:

- Uso do banco de dados
- Erros de autenticação
- Tentativas de acesso não autorizado

2. Use o painel do Netlify para monitorar:

- Disponibilidade do site
- Tempo de resposta
- Erros de implantação

## Manutenção

1. **Backups regulares:** Exporte periodicamente os dados do Firestore

2. **Atualizações de segurança:** Mantenha as bibliotecas e dependências atualizadas

3. **Revisão de código:** Realize auditorias periódicas para identificar vulnerabilidades

# Limites dos Planos Gratuitos

## Firebase (Plano Spark - Gratuito)

- Armazenamento: 1GB
- Transferência: 10GB/mês
- Operações de leitura: 50.000/dia
- Operações de escrita: 20.000/dia
- Operações de exclusão: 20.000/dia

Estes limites são mais que suficientes para o volume mencionado (50 requisições diárias e pico de 300 às sextas).

## Netlify (Plano Gratuito)

- Implantações: Ilimitadas
- Largura de banda: 100GB/mês
- Tempo de build: 300 minutos/mês
- Sites: Ilimitados
- Funções serverless: 125.000 invocações/mês

---

Este guia fornece todas as etapas necessárias para implantar o Sistema de Autorizações em um ambiente de produção seguro e sem custos. Seguindo estas instruções, o sistema estará pronto para uso com capacidade para lidar com o volume de requisições esperado.

# Sistema de Autorizações Digitais - SC Internacional

## Sobre a Refatoração

Este projeto passou por uma refatoração completa para melhorar sua estrutura, manutenibilidade e robustez. As principais mudanças incluem:

1. **Implementação do Padrão Module:** Todos os componentes principais agora seguem o padrão de design Module, encapsulando a lógica e expondo apenas as interfaces necessárias.
2. **Unificação de Arquivos com Funcionalidades Similares:** Arquivos que realizavam funções similares (como versões integradas e não integradas) foram unificados em controladores específicos.
3. **Serviço Centralizado de Autorizações:** Foi criado um serviço central (`autorizacao-service.js`) que gerencia todas as operações relacionadas às autorizações.
4. **Correção do Conversor de PDF:** O problema na conversão de arquivos Markdown para PDF foi corrigido, ajustando os caminhos relativos.

## Estrutura do Projeto

### Novos Arquivos

- `js/autorizacao-service.js`: Serviço central que gerencia todas as operações de autorizações.
- `js/solicitacao-controller.js`: Controlador unificado para as funcionalidades de solicitação.
- `js/supervisor-controller.js`: Controlador unificado para as funcionalidades do supervisor.
- `js/servico-social-controller.js`: Controlador unificado para as funcionalidades do serviço social.
- `js/monitor-controller.js`: Controlador unificado para as funcionalidades do monitor.
- `js/consulta-controller.js`: Controlador unificado para as funcionalidades de consulta.

# Como Usar

## Incluindo os Novos Scripts

Para utilizar os novos controladores, adicione as seguintes tags de script nas páginas HTML correspondentes:

```
<!-- Serviço de Autorização (deve ser incluído em todas as páginas) -->
<script src="js/autorizacao-service.js"></script>

<!-- Controladores específicos (incluir conforme necessário) -->
<script src="js/solicitacao-controller.js"></script> <!-- Para páginas de solicitação -->
<script src="js/supervisor-controller.js"></script> <!-- Para páginas do supervisor -->
<script src="js/servico-social-controller.js"></script> <!-- Para páginas do serviço social -->
<script src="js/monitor-controller.js"></script> <!-- Para páginas do monitor -->
<script src="js/consulta-controller.js"></script> <!-- Para páginas de consulta -->
```

## Fluxo de Funcionamento

### 1. Solicitação de Autorização:

- O atleta preenche o formulário de solicitação
- O sistema registra a solicitação e notifica o supervisor

### 2. Análise do Supervisor:

- O supervisor analisa a solicitação e aprova ou reprova
- Se aprovada, o sistema notifica o serviço social

### 3. Análise do Serviço Social:

- O serviço social analisa a solicitação e aprova ou reprova
- O sistema atualiza o status final da solicitação

### 4. Monitoramento:

- O monitor pode visualizar todas as solicitações e seus status

### 5. Consulta:

- O atleta pode consultar o status de sua solicitação usando o código fornecido

## Benefícios da Refatoração

- Código mais Limpo e Organizado:** A separação de responsabilidades torna o código mais fácil de entender e manter.
- Redução de Duplicação:** A lógica comum foi centralizada, eliminando duplicações.
- Facilidade de Manutenção:** Alterações futuras serão mais simples de implementar.
- Melhor Gerenciamento de Estado:** O estado da aplicação é gerenciado de forma mais consistente.

5. **Melhor Tratamento de Erros:** Implementação de validações e tratamento de erros mais robustos.

## Próximos Passos

1. Implementar testes automatizados
2. Migrar o armazenamento de dados do localStorage para um banco de dados
3. Implementar autenticação e autorização mais robustas
4. Melhorar a interface do usuário com feedback mais detalhado

# Fluxograma do Sistema de Autorizações Digitais

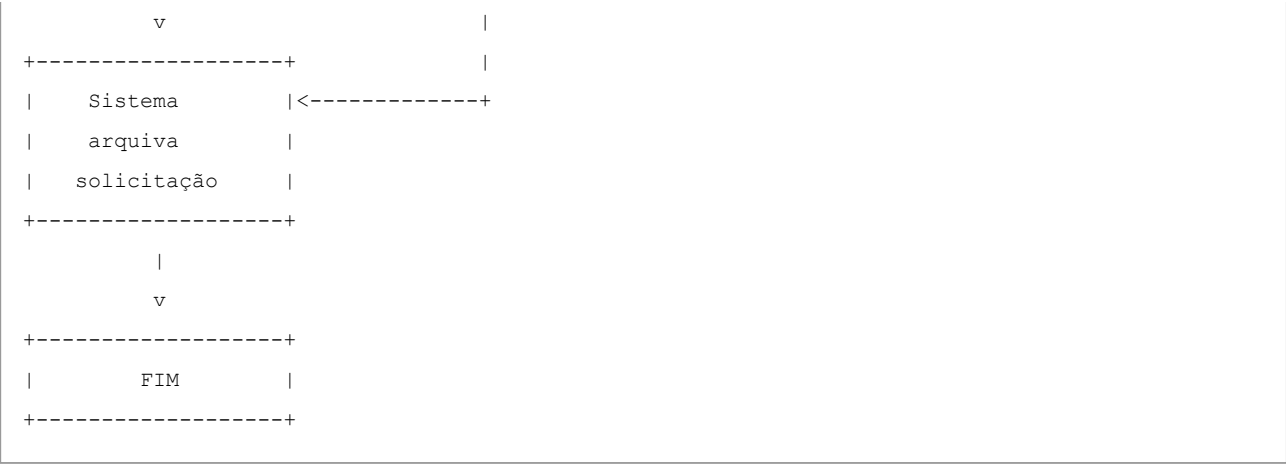
## Fluxo Completo do Processo

```

+-----+
|      INÍCIO      |
+-----+
|
|      v
+-----+
|  Atleta preenche  |
|  formulário de   |
|  solicitação     |
+-----+
|
|      v
+-----+      +-----+
| Sistema valida   |---->| Exibe mensagem  |
| os dados?       |Não | de erro      |
+-----+      +-----+
|
|      |
| Sim   |
|      |
|      v      |
+-----+      |
| Sistema registra |      |
| a solicitação e  |      |
| gera código      |      |
+-----+      |
|
|      |
|      v      |
+-----+      |
| Sistema notifica |      |
| supervisor da    |      |
| categoria        |      |
+-----+      |
|
|      |
|      v      |
+-----+      +-----+
| Supervisor analisa---->| Supervisor      |
| a solicitação?  |Não | reprova      |
+-----+      +-----+
|
|      |
| Sim   |
|      |
|      v      |
+-----+      |
| Supervisor      |      |
| aprova          |      |
+-----+      |
|
|      |
|      v      |
+-----+      |
| Sistema notifica |      |

```

serviço social	
+-----+	
v	
+-----+	+-----+
Serviço social	---->  Serviço social
valida?	Não   reprova
+-----+	+-----+
Sim	
v	
+-----+	
Serviço social	
envia WhatsApp	
para responsável	
+-----+	
v	
+-----+	+-----+
Responsável	---->  Solicitação
confirma?	Não   reprovada
+-----+	+-----+
Sim	
v	
+-----+	
Solicitação	
aprovada	
+-----+	
v	
+-----+	
Sistema notifica	
atleta	
+-----+	
v	
+-----+	
Monitor registra	
saída do atleta	
+-----+	
v	
+-----+	
Monitor registra	
retorno do atleta	
+-----+	



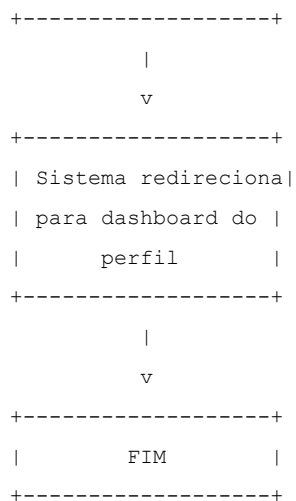
# Fluxo de Autenticação



```

+-----+
|      INÍCIO      |
+-----+
|
|      v
+-----+
|  Usuário acessa  |
|  página de login |
+-----+
|
|      v
+-----+
|  Usuário seleciona |
|      perfil      |
+-----+
|
|      v
+-----+      +-----+
| Primeiro acesso? |---->| Usuário insere   |
|                  |Sim  | credenciais padrão|
+-----+      +-----+
|                  |      |
|      Não        |      |
|      v          |      |
+-----+      +-----+
|  Usuário insere  |      | Sistema redireciona|
| credenciais      |      | para configuração |
| personalizadas   |      | de nova senha     |
+-----+      +-----+
|                  |      |
|      v          |      |
+-----+      +-----+
| Sistema valida   |      | Usuário define   |
| credenciais      |<----| nova senha     |
+-----+      +-----+
|
|      v
+-----+      +-----+
| Credenciais      |---->| Exibe mensagem |
| válidas?        |Não  | de erro         |
+-----+      +-----+
|
|      Sim
|      v
+-----+
| Sistema registra |
| informações do   |
| dispositivo      |

```



## Fluxo de Consulta de Status

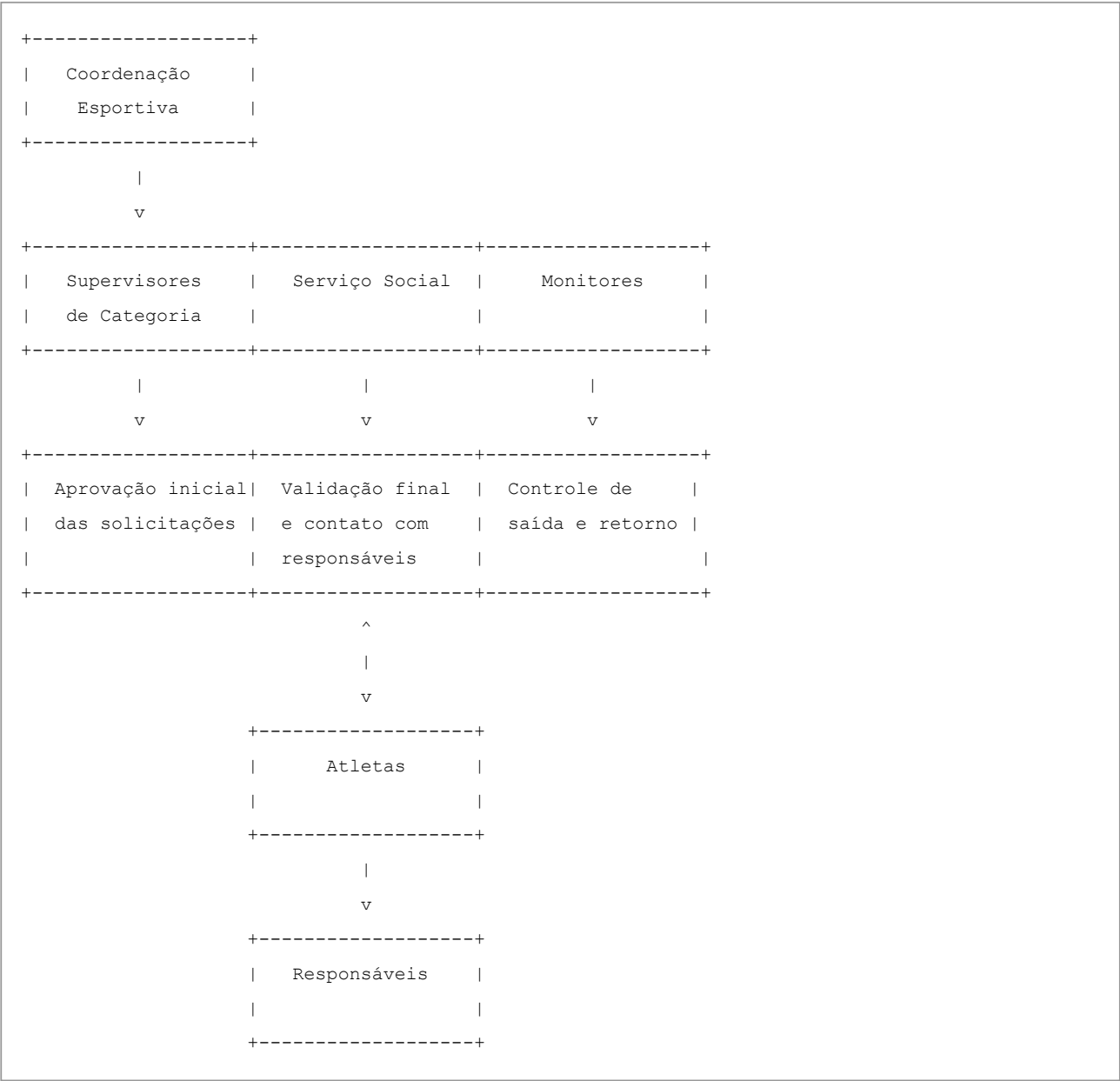
```

+-----+
|      INÍCIO      |
+-----+
|
|      v
+-----+
|  Atleta acessa   |
|  página de consulta
+-----+
|
|      v
+-----+      +-----+
| Atleta possui   |---->| Atleta insere   |
| código?         |Não  | dados para busca |
+-----+      +-----+
|
|      | Sim      |
|      v          |
+-----+      |
| Atleta insere   |
| código de       |<-----+
| solicitação     |
+-----+
|
|      v
+-----+      +-----+
| Sistema localiza |---->| Exibe mensagem  |
| solicitação?    |Não  | de não encontrado|
+-----+      +-----+
|
|      | Sim      |
|      v          |
+-----+
| Sistema exhibe  |
| detalhes da     |
| solicitação     |
+-----+
|
|      v
+-----+
| Sistema exhibe  |
| status atual e   |
| histórico        |
+-----+
|
|      v
+-----+

```



# Organograma Funcional



# Diagrama UML do Sistema de Autorizações Digitais

## Diagrama de Classes

-----+	-----+	-----+
Usuário	Solicitação	Notificação
-----+	-----+	-----+
- username	- id	- id
- password	- nome	- tipo
- perfil	- email	- categoria
- lastLogin	- data_nascimento	- id_solicitacao
- deviceInfo	- telefone	- nome_atleta
- passwordSet	- categoria	- data_envio
-----+	- data_saida	- lida
+ login()	- horario_saida	- mensagem
+ logout()	- data_retorno	-----+
+ resetPassword()	- horario_retorno	+ marcarComoLida()
-----+	- motivo_destino	+ enviar()
^	- nome_responsavel	-----+
	- telefone_responsavel	^
	- data_solicitacao	
-----+-----+	- status_supervisor	
	- status_servico_social	
-----+-----+	- status_final	
	- dispositivo	
	-----+	
	+ aprovar()	
	+ reprovar()	
	+ validar()	
	+ consultar()	
	-----+	
	^	
-----+-----+-----+-----+		
-----+-----+-----+-----+		
Atleta      Supervisor   Serviço      Monitor		
Social		
-----+-----+-----+-----+		
+ solicitar()	+ aprovar()	+ validar()
+ consultar()	+ reprovar()	+ reprovar()
		+ registrarRetorno()
-----+	-----+	-----+
-----+	-----+	
Confirmação	NotificacaoService	-----+
-----+	-----+	
- token	- notificacoes	
- id_solicitacao	-----+	
- nome_atleta	+ enviarNotificacaoSupervisor()	
- nome_responsavel	+ enviarNotificacaoServicoSocial()	

```

| - telefone_responsavel | + enviarNotificacaoAtleta() |
| - data_geracao         | + simularEnvioEmail() |
| - data_confirmacao     | + obterNotificacoesPendentes() |
| - ip_confirmacao       | + marcarComoLida() |
| - confirmado           | + salvarNotificacoes() |
| - hash_legal           | +-----+
+-----+ ^
| + confirmar()         | |
| + verificar()         | |
+-----+ |
      ^ |
      | |
      | |
+-----+ |
| ConfirmacaoService |-----+
+-----+
| - confirmacoes     |
+-----+
| + gerarLinkConfirmacao() |
| + gerarTokenUnico() |
| + registrarToken() |
| + processarConfirmacao() |
| + atualizarStatusSolicitacao() |
| + gerarHashLegal() |
| + salvarConfirmacoes() |
+-----+

```

## Diagrama de Sequência - Fluxo de Autorização

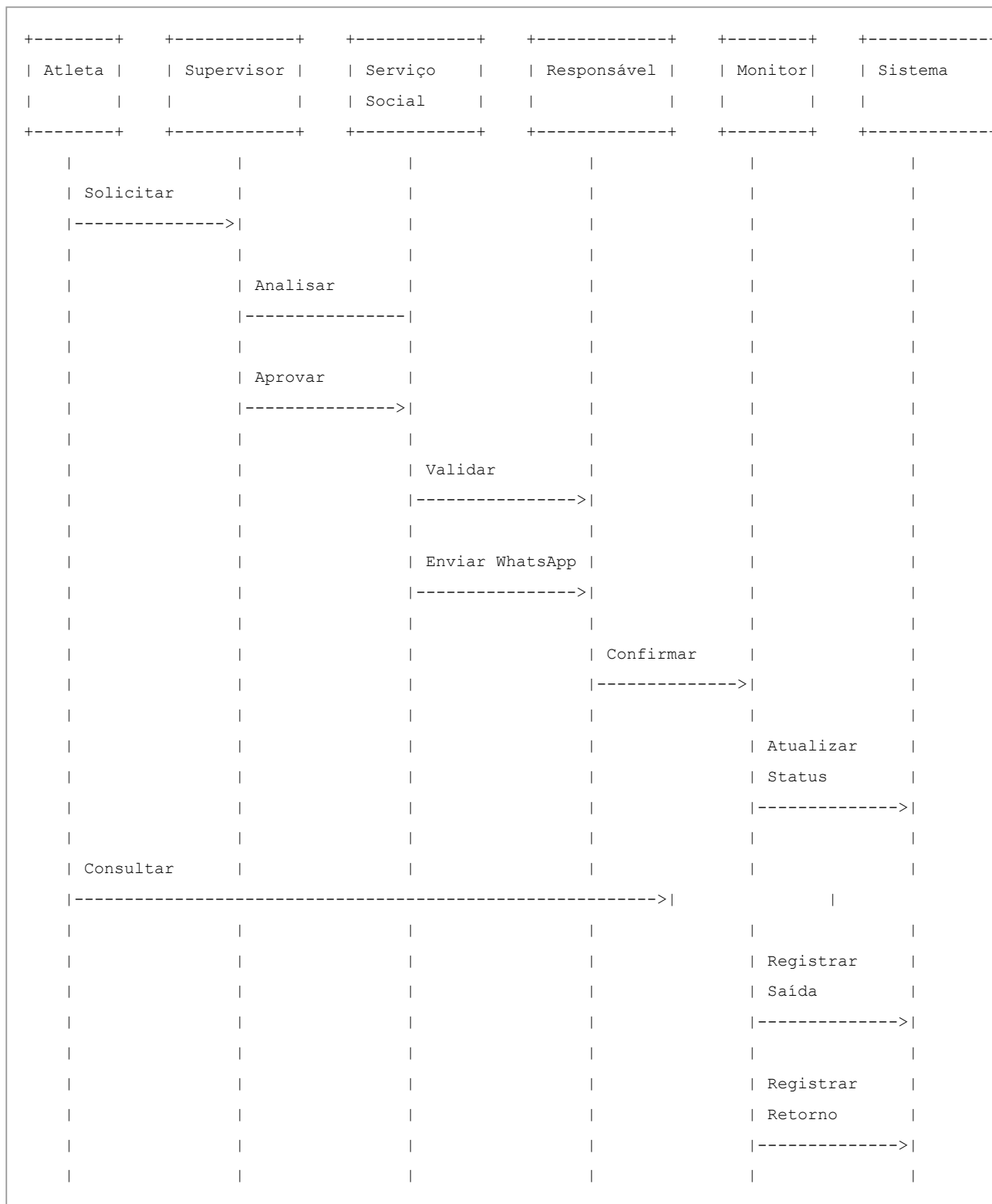
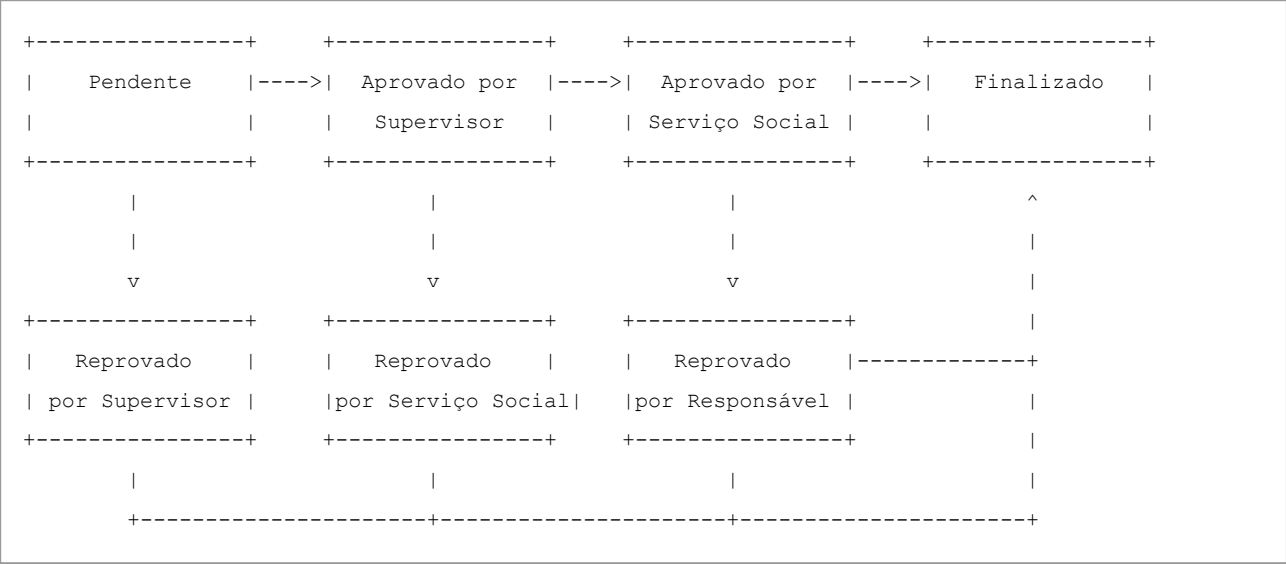
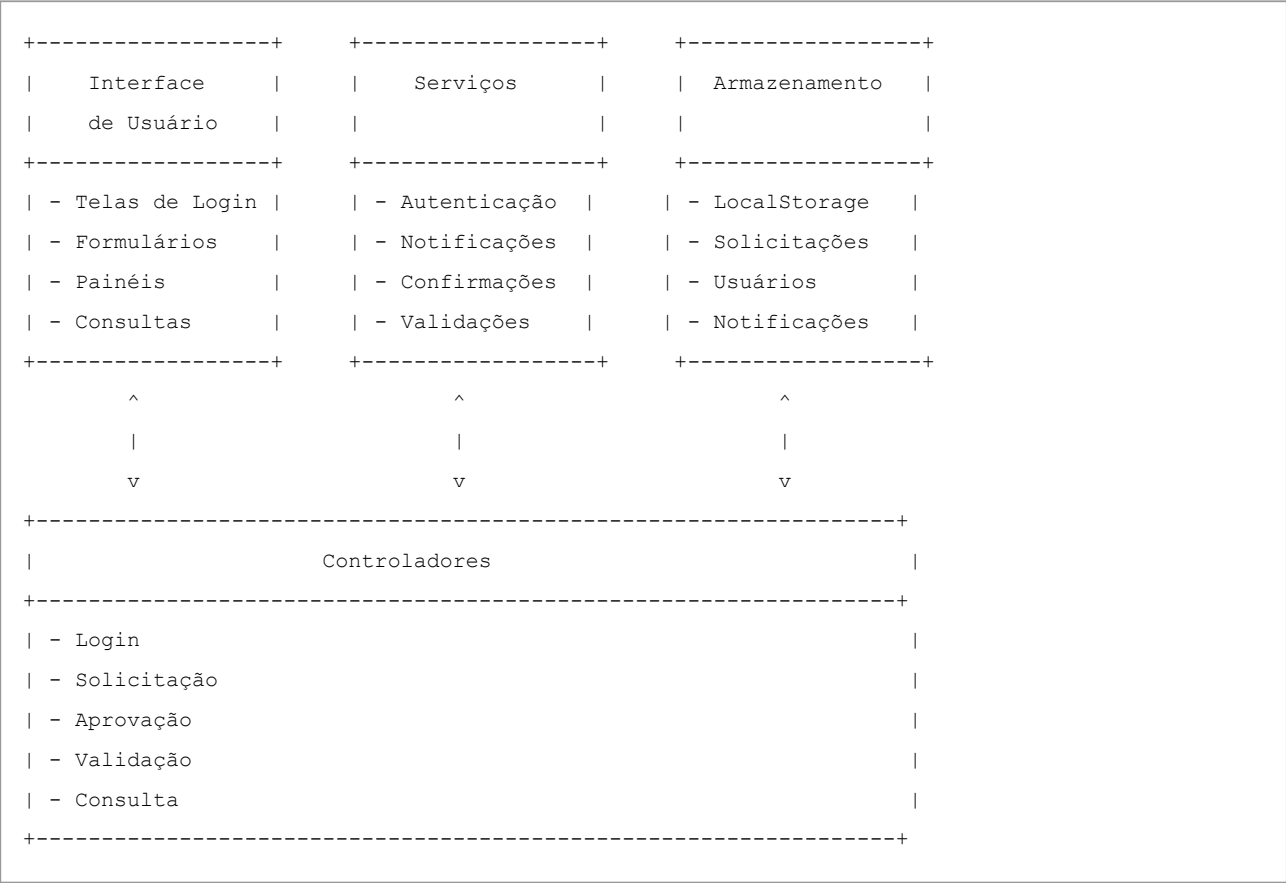


Diagrama de Estados - Ciclo de Vida da Solicitação



## Diagrama de Componentes



# Planejamento do Sistema de Autorizações Digitais v2



# Estrutura do Projeto

- ☒ Criar diretório do projeto
- ☒ Copiar imagem de fundo
- ☒ Definir estrutura de arquivos e diretórios
- ☒ Configurar ambiente de desenvolvimento

## Telas do Sistema

- ☒ Tela de login geral para atletas
- ☒ Formulário de solicitação de autorização
- ☒ Tela de consulta de status para atletas
- ☒ Painel de supervisores (por categoria)
- ☒ Painel de serviço social/monitores
- ☒ Tela de arquivamento e histórico

## Lógica de Backend

- ☒ Modelo de dados para autorizações
- ☒ Sistema de autenticação e permissões
- ☒ Lógica de aprovação em etapas
- ☒ Geração de IDs únicos e hashes
- ☒ Sistema de arquivamento com data/hora
- ☒ Integração com WhatsApp

## Notificações

- ☒ Notificação para supervisor quando atleta preenche formulário
- ☒ Notificação para serviço social quando supervisor aprova
- ☒ Mensagem para WhatsApp do responsável
- ☒ Confirmação de aprovação do responsável

## Validação e Testes

- ☒ Validar telas com o usuário
- ☒ Testar fluxo completo de aprovação
- ☒ Verificar conformidade com requisitos legais
- ☒ Testar arquivamento e recuperação de dados

## Entrega Final

- ☒ Documentação do sistema
- ☒ Manual de uso para cada perfil
- ☒ Implantação em ambiente de produção

