

# SISTEMA DE GESTÃO DE MATRÍCULAS

## DASHBOARD CRECHE ESTRELA DO ORIENTE

---

### Documentação Técnica Completa

**Projeto:** Dashboard Matrícula Creche

**Versão:** 1.0.0

**Data:** 07 de outubro de 2025

**Autor:** Pedro Oliveira

**Curso:** Desenvolvimento Web Avançado

**Status:** Aprovado

---

### CONTROLE DOCUMENTAL

Versão	Data	Autor	Mudanças
v1.0.0	07/10/2025	Pedro Oliveira	Versão inicial da documentação técnica
v0.9.0	06/10/2025	Pedro Oliveira	Correções de build e otimizações
v0.8.0	05/10/2025	Pedro Oliveira	Implementação completa do sistema

---

**Status do Documento:** Aprovado

**Última Revisão:** 07 de outubro de 2025

**Responsável pela Aprovação:** Pedro Oliveira

---

### SUMÁRIO

1. Visão, Propósito e Contexto
2. Convenções e Abreviações
3. Stakeholders e Personas
4. Escopo e Limites
5. Requisitos Funcionais
6. Requisitos Não Funcionais
7. Regras de Negócio
8. Casos de Uso e Fluxos
9. Modelo de Dados
10. Controles de Acesso (RBAC)
11. Arquitetura e Stack Tecnológico

## 1. VISÃO, PROPÓSITO E CONTEXTO

### 1.1 Problema Atual

As creches e instituições de educação infantil enfrentam desafios significativos na gestão manual de matrículas e informações dos alunos:

- **Processos manuais ineficientes:** Uso de planilhas e documentos físicos para controle de matrículas
- **Falta de centralização:** Informações dispersas em diferentes sistemas e arquivos
- **Dificuldade de acesso:** Professores e colaboradores sem acesso rápido aos dados dos alunos
- **Relatórios demorados:** Geração manual de relatórios para análises estatísticas
- **Risco de perda de dados:** Documentos físicos suscetíveis a perda ou danos

### 1.2 Objetivos do Sistema

1. **Digitalizar completamente** o processo de gestão de matrículas da creche
2. **Centralizar informações** de alunos em uma plataforma única e segura
3. **Facilitar o acesso** de professores e colaboradores aos dados necessários
4. **Automatizar a geração** de relatórios estatísticos e exportações
5. **Garantir segurança** e integridade dos dados com controle de acesso

### 1.3 Valor para Usuários e Organização

**Para Usuários:** - Acesso rápido e organizado às informações dos alunos - Interface intuitiva e responsiva para uso em qualquer dispositivo - Redução significativa do tempo gasto em tarefas administrativas

**Para a Organização:** - Maior eficiência operacional e redução de custos administrativos - Melhor controle e análise estatística do corpo discente - Conformidade com boas práticas de gestão educacional - Base sólida para expansão e escalabilidade futura

### 1.4 Escopo Tecnológico

**Stack Principal:** - **Frontend:** Next.js 15 + React 19 + TypeScript - **Estilização:** Tailwind CSS 4.0 - **Banco de Dados:** Firebase Firestore (NoSQL) - **Autenticação:** Firebase Authentication - **Hospedagem:** Vercel (recomendado) ou plataformas similares

**Serviços Integrados:** - Firebase Analytics para monitoramento - Validação com Zod + React Hook Form - Exportação CSV com PapaParse - Sistema de ícones Lucide React

---

## 2. CONVENÇÕES E ABREVIACÕES

### 2.1 Abreviações Técnicas

	Sigla	Significado
	RF	Requisito Funcional
	RNF	Requisito Não Funcional
	CRUD	Create, Read, Update, Delete
	CSV	Comma-Separated Values
	LGPD	Lei Geral de Proteção de Dados
	WCAG	Web Content Accessibility Guidelines
	RBAC	Role-Based Access Control
	MVP	Minimum Viable Product
	UI/UX	User Interface/User Experience
	API	Application Programming Interface

### 2.2 Convenções de Nomenclatura

**Coleções Firebase:** - **users:** Dados dos usuários do sistema - **students:** Informações dos alunos

**Padrões de ID:** - Requisitos Funcionais: RF-[DOMINIO]-[NUMERO] - Exemplo: RF-AUTH-001, RF-STUD-002 - Requisitos Não Funcionais: RNF-[CATEGORIA]-[NUMERO] - Exemplo: RNF-PERF-001, RNF-SEC-002

**Status e Estados:** - **active/inactive:** Status de alunos - **professor/colaborador:** Roles de usuários

---

## 3. STAKEHOLDERS E PERSONAS

### 3.1 Stakeholders

Stakeholder	Papel	Objetivo	Influência
Direção da Creche	Decisor	Eficiência operacional e controle administrativo	Alta

Stakeholder	Papel	Objetivo	Influência
Coordenação Pedagógica	Usuário Primário	Acesso rápido aos dados dos alunos	Alta
Professores	Usuário Final	Consulta de informações para planejamento de atividades	Média
Secretaria	Usuário Administrativo	Gestão de matrículas e relatórios	Alta
Responsáveis pelos Alunos	Beneficiário Indireto	Dados atualizados e processos ágeis	Baixa

### 3.2 Personas

#### Persona 1: Maria Santos - Coordenadora Pedagógica

- **Idade:** 45 anos
- **Experiência:** 20 anos em educação infantil
- **Metas:** Acessar rapidamente informações médicas e de emergência dos alunos
- **Dores:** Perda de tempo procurando documentos físicos
- **Critérios de Sucesso:** Encontrar qualquer informação de aluno em menos de 30 segundos

#### Persona 2: João Silva - Professor

- **Idade:** 32 anos
- **Experiência:** 8 anos como educador
- **Metas:** Consultar alergias e informações especiais antes das atividades
- **Dores:** Não ter acesso fácil aos dados durante as aulas
- **Critérios de Sucesso:** Interface simples que funcione no tablet da sala

#### Persona 3: Ana Costa - Secretária Administrativa

- **Idade:** 38 anos
- **Experiência:** 12 anos em administração escolar
- **Metas:** Realizar matrículas rapidamente e gerar relatórios mensais
- **Dores:** Retrabalho na digitação e dificuldade para gerar estatísticas
- **Critérios de Sucesso:** Reduzir tempo de matrícula de 30 para 10 minutos

#### Persona 4: Carlos Roberto - Diretor

- **Idade:** 52 anos
  - **Experiência:** 25 anos em gestão educacional
  - **Metas:** Ter visão estatística do crescimento e análises estratégicas
  - **Dores:** Relatórios demorados e dados desatualizados
  - **Critérios de Sucesso:** Dashboards atualizados em tempo real
- 

## 4. ESCOPO E LIMITES

### 4.1 Funcionalidades IN SCOPE (MVP)

**Módulo de Autenticação:** - Login seguro com email/senha - Controle de sessão e logout - Recuperação de senha

**Módulo de Gestão de Alunos:** - Cadastro completo de novos alunos - Listagem com filtros avançados - Edição de informações existentes - Consulta detalhada de dados

**Módulo de Relatórios:** - Dashboard com estatísticas gerais - Exportação em CSV por diferentes critérios - Relatórios de matrículas recentes - Distribuição por faixa etária

**Interface e Usabilidade:** - Design responsivo (mobile-first) - Interface intuitiva e acessível - Feedback visual para todas as ações

### 4.2 Funcionalidades OUT OF SCOPE

- Módulo financeiro (mensalidades, pagamentos)
- Sistema de comunicação (mensagens, avisos)
- Agenda de eventos e atividades
- Controle de presença/frequência
- Sistema de notas e avaliações
- Integração com sistemas terceiros
- Aplicativo mobile nativo
- Funcionalidades offline

### 4.3 Assunções

- Usuários têm acesso à internet estável
- Dispositivos suportam navegadores modernos
- Equipe tem conhecimento básico em tecnologia
- Firebase mantém disponibilidade adequada

### 4.4 Dependências

- Conta Google Cloud/Firebase ativa
- Domínio para hospedagem (opcional)

- Certificado SSL para produção
- Backup de dados implementado

#### 4.5 Restrições

**Técnicas:** - Limite de 1GB de armazenamento Firebase (plano gratuito) - Máximo 50.000 leituras/dia no Firestore - Compatibilidade com Chrome 90+, Firefox 88+, Safari 14+

**Legais:** - Conformidade com LGPD - Proteção de dados de menores - Termo de uso e política de privacidade

**Orçamentárias:** - Desenvolvimento em plano gratuito do Firebase - Hospedagem gratuita na Vercel - Sem custos de licenças adicionais

---

## 5. REQUISITOS FUNCIONAIS

### 5.1 Domínio de Autenticação

#### RF-AUTH-001: Login do Sistema

- **Descrição:** O sistema deve permitir login de usuários cadastrados
- **Ator:** Professor, Colaborador
- **Prioridade:** Alta
- **Gatilho:** Usuário acessa a URL principal
- **Entradas:** Email, senha
- **Saídas:** Redirecionamento para dashboard ou mensagem de erro
- **Regras:** Email deve ser válido, senha mínimo 6 caracteres
- **Erro:** “Email ou senha incorretos. Tente novamente.”

#### RF-AUTH-002: Logout do Sistema

- **Descrição:** O sistema deve permitir logout seguro
- **Ator:** Usuário logado
- **Prioridade:** Alta
- **Gatilho:** Clique no botão logout
- **Entradas:** Ação do usuário
- **Saídas:** Redirecionamento para tela de login
- **Regras:** Limpar sessão e cache local

#### RF-AUTH-003: Controle de Sessão

- **Descrição:** O sistema deve manter sessão ativa e validar permissões
- **Ator:** Usuário logado
- **Prioridade:** Alta
- **Gatilho:** Navegação entre páginas
- **Entradas:** Token de sessão

- **Saídas:** Acesso permitido ou redirecionamento
- **Regras:** Sessão expira em 24 horas de inatividade

## 5.2 Domínio de Gestão de Alunos

### RF-STUD-001: Cadastro de Aluno

- **Descrição:** O sistema deve permitir cadastro completo de novos alunos
- **Ator:** Colaborador, Professor
- **Prioridade:** Alta
- **Gatilho:** Clique em “Cadastrar Aluno”
- **Entradas:** Nome, data nascimento, dados do responsável, informações médicas
- **Saídas:** Confirmação de sucesso ou mensagens de erro
- **Regras:** Todos os campos obrigatórios devem ser preenchidos
- **Erro:** “Campo [nome] é obrigatório”

### RF-STUD-002: Listagem de Alunos

- **Descrição:** O sistema deve exibir lista paginada de alunos cadastrados
- **Ator:** Professor, Colaborador
- **Prioridade:** Alta
- **Gatilho:** Acesso à aba “Alunos”
- **Entradas:** Filtros opcionais (nome, idade, status)
- **Saídas:** Lista formatada com informações principais
- **Regras:** Ordenação alfabética por nome

### RF-STUD-003: Busca e Filtros

- **Descrição:** O sistema deve permitir busca por nome e filtros por idade/status
- **Ator:** Professor, Colaborador
- **Prioridade:** Média
- **Gatilho:** Digitação no campo de busca ou seleção de filtro
- **Entradas:** Termo de busca, filtros selecionados
- **Saídas:** Lista filtrada em tempo real
- **Regras:** Busca case-insensitive, filtros cumulativos

### RF-STUD-004: Edição de Aluno

- **Descrição:** O sistema deve permitir edição de dados de alunos existentes
- **Ator:** Colaborador
- **Prioridade:** Alta
- **Gatilho:** Clique em “Editar” na listagem
- **Entradas:** Dados modificados do aluno
- **Saídas:** Confirmação de atualização
- **Regras:** Validações iguais ao cadastro

#### **RF-STUD-005: Visualização Detalhada**

- **Descrição:** O sistema deve exibir todos os dados de um aluno específico
- **Ator:** Professor, Colaborador
- **Prioridade:** Média
- **Gatilho:** Clique no nome do aluno
- **Entradas:** ID do aluno
- **Saídas:** Modal ou página com informações completas
- **Regras:** Exibir histórico de alterações

### **5.3 Domínio de Relatórios**

#### **RF-REP-001: Dashboard Estatístico**

- **Descrição:** O sistema deve exibir estatísticas gerais em dashboard
- **Ator:** Professor, Colaborador
- **Prioridade:** Média
- **Gatilho:** Acesso à aba “Relatórios”
- **Entradas:** Data atual
- **Saídas:** Cards com números atualizados
- **Regras:** Atualização em tempo real

#### **RF-REP-002: Exportação CSV**

- **Descrição:** O sistema deve permitir exportação de dados em CSV
- **Ator:** Colaborador
- **Prioridade:** Alta
- **Gatilho:** Clique em botões de exportação
- **Entradas:** Critério de exportação (todos, ativos, por idade)
- **Saídas:** Download automático do arquivo CSV
- **Regras:** Nome do arquivo com timestamp

#### **RF-REP-003: Relatório de Matrículas Recentes**

- **Descrição:** O sistema deve exibir alunos matriculados nos últimos 30 dias
- **Ator:** Colaborador
- **Prioridade:** Baixa
- **Gatilho:** Visualização automática no dashboard
- **Entradas:** Data atual
- **Saídas:** Tabela com matrículas recentes
- **Regras:** Ordenação por data de matrícula decrescente

---

## **6. REQUISITOS NÃO FUNCIONAIS**

### **6.1 Performance**

#### **RNF-PERF-001: Tempo de Carregamento**



- **Métrica:** Página inicial deve carregar em menos de 2 segundos
- **Ambiente:** Conexão 3G (1.6 Mbps)
- **Medição:** Core Web Vitals, Lighthouse

#### **RNF-PERF-002: Tempo de Resposta**

- **Métrica:** Operações CRUD devem responder em menos de 1 segundo
- **Ambiente:** Condições normais de rede
- **Medição:** Network tab do navegador

#### **RNF-PERF-003: Throughput**

- **Métrica:** Sistema deve suportar 50 usuários simultâneos
- **Ambiente:** Produção
- **Medição:** Testes de carga com Artillery ou similares

### **6.2 Segurança e Privacidade**

#### **RNF-SEC-001: Autenticação**

- **Requisito:** Autenticação obrigatória para todas as funcionalidades
- **Implementação:** Firebase Authentication com email/senha
- **Validação:** Teste de acesso sem login deve falhar

#### **RNF-SEC-002: Proteção de Dados (LGPD)**

- **Requisito:** Criptografia de dados sensíveis e logs sem PII
- **Implementação:** Firestore Security Rules, HTTPS obrigatório
- **Validação:** Auditoria de segurança e privacy by design

#### **RNF-SEC-003: Controle de Acesso**

- **Requisito:** Usuários só acessam dados permitidos pelo seu role
- **Implementação:** RBAC com validação client e server-side
- **Validação:** Testes de permissões cruzadas

#### **RNF-SEC-004: Sessão Segura**

- **Requisito:** Sessões expiram automaticamente após inatividade
- **Implementação:** Token refresh do Firebase Auth
- **Validação:** Teste de timeout de sessão

### **6.3 Acessibilidade**

#### **RNF-ACC-001: Conformidade WCAG 2.2 AA**

- **Requisito:** Interface acessível para usuários com deficiência
- **Implementação:** Semantic HTML, ARIA labels, contraste adequado
- **Validação:** Testes com screen readers e ferramentas automatizadas

#### **RNF-ACC-002: Navegação por Teclado**

- **Requisito:** Todas as funcionalidades acessíveis via teclado
- **Implementação:** Focus management, tab order correto
- **Validação:** Teste manual de navegação sem mouse

### **6.4 Confiabilidade e Observabilidade**

#### **RNF-REL-001: Disponibilidade**

- **Requisito:** 99.5% de uptime mensal
- **Implementação:** Hospedagem em Vercel + Firebase
- **Medição:** Monitoring com Firebase Analytics

#### **RNF-REL-002: Backup de Dados**

- **Requisito:** Backup automático diário dos dados
- **Implementação:** Firebase Firestore backup nativo
- **Validação:** Testes de restauração mensais

#### **RNF-REL-003: Monitoramento**

- **Requisito:** Logs de erros e métricas de uso
- **Implementação:** Firebase Analytics + Console logs
- **Validação:** Dashboard de métricas funcionando

#### **RNF-REL-004: Recuperação de Erros**

- **Requisito:** Sistema deve se recuperar graciosamente de falhas
- **Implementação:** Try-catch blocks, fallbacks, loading states
- **Validação:** Testes de cenários de erro

### **6.5 Usabilidade**

#### **RNF-USA-001: Design Responsivo**

- **Requisito:** Interface adaptável para mobile, tablet e desktop
- **Implementação:** Tailwind CSS mobile-first approach
- **Validação:** Testes em dispositivos múltiplos

#### **RNF-USA-002: Consistência Visual**

- **Requisito:** Componentes visuais padronizados
- **Implementação:** Design system com Tailwind CSS
- **Validação:** Review visual de todas as telas

#### **RNF-USA-003: Feedback do Usuário**

- **Requisito:** Feedback claro para todas as ações
- **Implementação:** Loading states, mensagens de sucesso/erro, toasts
- **Validação:** Teste de usabilidade com usuários reais

#### **RNF-USA-004: Curva de Aprendizado**

- **Requisito:** Usuários devem conseguir usar o sistema sem treinamento
  - **Implementação:** Interface intuitiva, labels claros, navegação óbvia
  - **Validação:** Testes de usabilidade com novos usuários
- 

## **7. REGRAS DE NEGÓCIO**

### **7.1 Regras de Cadastro de Alunos**

#### **RN-001: Validação de Idade**

- **Regra:** Alunos devem ter entre 6 meses e 6 anos
- **Cálculo:** `idade = ano_atual - ano_nascimento - (se ainda não fez aniversário este ano ? 1 : 0)`
- **Origem:** Regulamentação da creche
- **Exceção:** Casos especiais aprovados pela direção

#### **RN-002: Dados Obrigatórios**

- **Regra:** Campos obrigatórios: nome, data nascimento, responsável, telefone, endereço
- **Validação:** Sistema não permite salvar sem preenchimento completo
- **Origem:** Norma interna da instituição
- **Exceção:** Nenhuma

#### **RN-003: Formato de Telefone**

- **Regra:** Telefones devem ter formato brasileiro (11) 99999-9999
- **Validação:** Regex: `/^\(\d{2}\)\s\d{4,5}-\d{4}$/`
- **Origem:** Padrão nacional de telecomunicações
- **Exceção:** Telefones internacionais para emergência

#### **RN-004: Email Único por Responsável**

- **Regra:** Cada email de responsável pode estar associado a múltiplos alunos
- **Validação:** Permite emails duplicados (irmãos)
- **Origem:** Prática comum em famílias
- **Exceção:** Nenhuma

## 7.2 Regras de Status e Estados

### RN-005: Status de Aluno

- **Regra:** Alunos podem estar “ativo” ou “inativo”
- **Estados:**
  - **active:** Frequentando regularmente
  - **inactive:** Matrícula suspensa ou cancelada
- **Origem:** Controle administrativo
- **Exceção:** Status “transferido” considerado como “inactive”

### RN-006: Cálculo de Idade Dinâmico

- **Regra:** Idade é calculada automaticamente com base na data atual
- **Cálculo:** Recalculado a cada visualização
- **Origem:** Necessidade de informação sempre atualizada
- **Exceção:** Nenhuma

## 7.3 Regras de Acesso e Permissões

### RN-007: Hierarquia de Usuários

- **Regra:** colaborador tem acesso completo, professor tem acesso limitado
- **Permissões:**
  - Colaborador: CRUD completo em alunos
  - Professor: Somente leitura e consulta
- **Origem:** Organograma da instituição
- **Exceção:** Professores podem ter permissão especial por período

### RN-008: Restrição de Edição

- **Regra:** Apenas colaboradores podem editar dados de alunos
- **Validação:** Verificação de role antes de exibir botões de edição
- **Origem:** Política de segurança de dados
- **Exceção:** Professores podem sugerir correções via solicitação

## 7.4 Regras de Relatórios

### RN-009: Critério de Matrícula Recente

- **Regra:** Matrículas recentes são dos últimos 30 dias
- **Cálculo:**  $\text{data\_matricula} \geq \text{data\_atual} - 30 \text{ dias}$
- **Origem:** Definição administrativa para acompanhamento
- **Exceção:** Período pode ser alterado sazonalmente

### RN-010: Agrupamento por Idade

- **Regra:** Faixas etárias: 0-1, 2-3, 4-5, 6+ anos

- **Cálculo:** Baseado na idade calculada dinamicamente
- **Origem:** Divisão pedagógica da instituição
- **Exceção:** Agrupamentos especiais para eventos

## 7.5 Regras de Validação de Dados

### RN-011: Formato de Data

- **Regra:** Datas devem estar no formato DD/MM/AAAA para exibição
- **Validação:** Conversão automática de ISO para formato brasileiro
- **Origem:** Padrão nacional
- **Exceção:** APIs internas usam ISO 8601

### RN-012: Campos Opcionais

- **Regra:** Informações médicas e alergias são opcionais mas recomendadas
- **Validação:** Sistema aceita valores vazios
- **Origem:** Algumas famílias podem não ter informações específicas
- **Exceção:** Casos com necessidades especiais devem preencher

---

## 8. CASOS DE USO E FLUXOS

### 8.1 UC-001: Realizar Login

**Ator Primário:** Professor/Colaborador

**Objetivo:** Autenticar-se no sistema para acessar funcionalidades

**Pré-condições:** Usuário possui conta cadastrada

**Pós-condições:** Usuário logado e redirecionado para dashboard

**Fluxo Principal:** 1. Usuário acessa a URL do sistema 2. Sistema exibe tela de login 3. Usuário preenche email e senha 4. Usuário clica em “Entrar” 5. Sistema valida credenciais no Firebase Auth 6. Sistema busca dados do usuário no Firestore 7. Sistema cria sessão e redireciona para dashboard

**Fluxos Alternativos:** - **5a.** Credenciais inválidas: Sistema exibe “Email ou senha incorretos” - **6a.** Usuário não encontrado no Firestore: Sistema cria documento automaticamente

**Fluxos de Exceção:** - **5e.** Erro de rede: Sistema exibe “Erro de conexão. Tente novamente” - **6e.** Firestore indisponível: Sistema permite login mas com funcionalidades limitadas

### 8.2 UC-002: Cadastrar Novo Aluno

**Ator Primário:** Colaborador

**Objetivo:** Adicionar novo aluno ao sistema

**Pré-condições:** Usuário logado com perfil de colaborador

**Pós-condições:** Aluno salvo no banco de dados

**Fluxo Principal:** 1. Usuário acessa aba “Cadastrar” 2. Sistema exibe formulário de cadastro 3. Usuário preenche dados obrigatórios do aluno 4. Usuário preenche dados do responsável 5. Usuário preenche informações adicionais (opcional) 6. Usuário clica em “Cadastrar Aluno” 7. Sistema valida todos os campos 8. Sistema calcula idade automaticamente 9. Sistema salva no Firestore 10. Sistema exibe mensagem de sucesso 11. Sistema limpa formulário para novo cadastro

**Fluxos Alternativos:** - **7a.** Campos obrigatórios vazios: Sistema destaca campos e impede envio - **7b.** Email inválido: Sistema exibe “Formato de email inválido” - **7c.** Telefone inválido: Sistema exibe “Formato de telefone inválido”

**Fluxos de Exceção:** - **9e.** Erro no Firestore: Sistema exibe “Erro ao salvar. Tente novamente” - **8e.** Erro de cálculo: Sistema usa data padrão e notifica administrador

### 8.3 UC-003: Buscar e Filtrar Alunos

**Ator Primário:** Professor/Colaborador

**Objetivo:** Encontrar alunos específicos na listagem

**Pré-condições:** Usuário logado, existem alunos cadastrados

**Pós-condições:** Lista filtrada exibida

**Fluxo Principal:** 1. Usuário acessa aba “Alunos” 2. Sistema carrega lista completa de alunos 3. Usuário digita nome no campo de busca 4. Sistema filtra lista em tempo real 5. Usuário seleciona filtro de idade (opcional) 6. Sistema aplica filtro adicional 7. Usuário seleciona filtro de status (opcional) 8. Sistema exibe lista final filtrada

**Fluxos Alternativos:** - **4a.** Nenhum resultado encontrado: Sistema exibe “Nenhum aluno encontrado” - **3a.** Usuário limpa busca: Sistema restaura lista completa

**Fluxos de Exceção:** - **2e.** Erro ao carregar: Sistema exibe loading e retry - **4e.** Erro de filtro: Sistema ignora filtro inválido

### 8.4 UC-004: Editar Dados do Aluno

**Ator Primário:** Colaborador

**Objetivo:** Atualizar informações de aluno existente

**Pré-condições:** Usuário logado como colaborador, aluno existe

**Pós-condições:** Dados atualizados no banco

**Fluxo Principal:** 1. Usuário acessa lista de alunos 2. Usuário clica em “Editar” de um aluno específico 3. Sistema abre modal de edição 4. Sistema pré-preenche formulário com dados atuais 5. Usuário modifica campos desejados 6. Usuário clica em “Salvar Alterações” 7. Sistema valida mudanças 8. Sistema atualiza registro no Firestore 9. Sistema fecha modal e atualiza lista 10. Sistema exibe mensagem de sucesso

**Fluxos Alternativos:** - **3a.** Usuário não tem permissão: Sistema não exibe botão editar - **5a.** Usuário cancela edição: Sistema fecha modal sem salvar

**Fluxos de Exceção:** - **8e.** Erro ao salvar: Sistema mantém modal aberto e exibe erro - **4e.** Erro ao carregar dados: Sistema exibe formulário vazio com aviso

## 8.5 UC-005: Gerar Relatório CSV

**Ator Primário:** Colaborador

**Objetivo:** Exportar dados de alunos em formato CSV

**Pré-condições:** Usuário logado, existem alunos cadastrados

**Pós-condições:** Arquivo CSV baixado

**Fluxo Principal:** 1. Usuário acessa aba “Relatórios” 2. Sistema exibe dashboard com estatísticas 3. Usuário seleciona tipo de relatório desejado 4. Usuário clica no botão de exportação correspondente 5. Sistema filtra dados conforme critério 6. Sistema formata dados para CSV 7. Sistema gera arquivo com timestamp 8. Sistema inicia download automaticamente

**Fluxos Alternativos:** - **5a.** Nenhum dado para critério: Sistema exibe “Nenhum dado encontrado” - **3a.** Usuário escolhe “Todos os alunos”: Sistema exporta base completa

**Fluxos de Exceção:** - **6e.** Erro na formatação: Sistema tenta formato simplificado - **8e.** Erro no download: Sistema oferece link direto para arquivo

---

## 9. MODELO DE DADOS

### 9.1 Entidade: Users

**Coleção:** users

**Descrição:** Dados dos usuários do sistema (professores e colaboradores)

Campo	Tipo	Obrigatório	Descrição	Validação
id	string	Sim	ID único do usuário (Firebase UID)	UUID Firebase
email	string	Sim	Email para login	Formato email válido
name	string	Sim	Nome completo do usuário	Mínimo 2 caracteres
role	enum	Sim	Perfil de acesso	‘professor’ ou ‘colaborador’
createdAt	Timestamp	Sim	Data de criação do registro	Data válida

**Índices:** - **email** (único) - **role** (composto com createdAt para consultas administrativas)

**Regras de Integridade:** - Email deve ser único no sistema - Role define permissões de acesso - CreatedAt é definido automaticamente no cadastro

## 9.2 Entidade: Students

**Coleção:** students

**Descrição:** Informações completas dos alunos matriculados

Campo	Tipo	Obrigatório	Descrição	Validação
id	string	Sim	ID único do aluno	UUID Firestore
name	string	Sim	Nome completo do aluno	Mínimo 2 caracteres
birthDate	Timestamp	Sim	Data de nascimento	Data válida no passado
age	number	Sim	Idade calculada em anos	Calculado automaticamente
parentName	string	Sim	Nome do responsável	Mínimo 2 caracteres
parentPhone	string	Sim	Telefone do responsável	Formato brasileiro
parentEmail	string	Sim	Email do responsável	Formato email válido
address	string	Sim	Endereço residencial	Mínimo 10 caracteres
medicalInfo	string	Não	Informações médicas relevantes	Texto livre
allergies	array	Não	Lista de alergias conhecidas	Array de strings
emergencyContact	string	Sim	Nome do contato de emergência	Mínimo 2 caracteres
emergencyPhone	string	Sim	Telefone de emergência	Formato brasileiro
enrollmentDate	Timestamp	Sim	Data de matrícula	Data válida
status	enum	Sim	Status do aluno	'active' ou 'inactive'
createdAt	Timestamp	Sim	Data de criação	Data automática



Campo	Tipo	Obrigatório	Descrição	Validação
updatedAt	Timestamp	Sim	Data da última alteração	Data automática

**Índices:** - `name` (para busca por nome) - `status` (para filtros rápidos) - `enrollmentDate` (para relatórios cronológicos) - `age` (para agrupamentos etários)

**Relacionamentos:** - Não há relacionamentos diretos entre coleções - `createdBy` e `updatedAt` poderiam referenciar `users` (futuro)

**Regras de Validação por Campo:** - `birthDate`: Deve ser posterior a 01/01/2018 e anterior à data atual - `age`: Recalculado automaticamente baseado em `birthDate` - `parentPhone` e `emergencyPhone`: Regex `/^\(\d{2}\)\s\d{4,5}-\d{4}$/` - `parentEmail`: Formato RFC 5322 compliant - `allergies`: Máximo 10 itens, cada item máximo 50 caracteres - `enrollmentDate`: Não pode ser futura

### 9.3 Consultas e Performance

#### Consultas Principais:

```
// Listar alunos ativos ordenados por nome
students.where('status', '==', 'active').orderBy('name')

// Buscar por nome (case insensitive requer índice especial)
students.where('name', '>=', searchTerm).where('name', '<=', searchTerm + '\uf8ff')

// Alunos por faixa etária
students.where('age', '>=', minAge).where('age', '<=', maxAge)

// Matrículas recentes (últimos 30 dias)
students.where('enrollmentDate', '>=', thirtyDaysAgo).orderBy('enrollmentDate', 'desc')
```

**Requisitos de Exportação:** - CSV deve incluir todos os campos visíveis ao usuário - Formato de datas em DD/MM/AAAA para compatibilidade brasileira - Campos opcionais vazios devem aparecer como string vazia - Encoding UTF-8 para caracteres especiais

**Estratégia de Backup:** - Firebase Firestore oferece backup automático - Exportação manual mensal em formato JSON - Retenção de 6 meses de backups históricos

## 10. CONTROLES DE ACESSO (RBAC)

### 10.1 Perfis de Usuário

#### Perfil: COLABORADOR

- **Descrição:** Funcionários administrativos com acesso completo
- **Responsabilidades:** Gestão completa de matrículas e relatórios
- **Quantidade Esperada:** 2-3 usuários

#### Perfil: PROFESSOR

- **Descrição:** Educadores com acesso de consulta
- **Responsabilidades:** Consulta de informações para atividades pedagógicas
- **Quantidade Esperada:** 8-12 usuários

### 10.2 Matriz de Permissões

Ação / Recurso	COLABORADOR	PROFESSOR
<b>Autenticação</b>		
Login		
Logout		
<b>Alunos - Consulta</b>		
Listar alunos		
Buscar/Filtrar		
Ver detalhes		
<b>Alunos - Gestão</b>		
Cadastrar novo		
Editar existente		
Alterar status		
<b>Relatórios</b>		
Ver dashboard		
Exportar CSV		
Relatórios detalhados		

**Legenda:** - Permitido - Negado

### 10.3 Implementação Técnica

#### Frontend (React/Next.js)

```
// Hook para verificação de permissões
const { user } = useAuth();
const canEdit = user?.role === 'colaborador';

// Renderização condicional
```

```
{canEdit && (
  <button onClick={handleEdit}>Editar</button>
)}
```

## Backend (Firebase Security Rules)

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // Usuários podem ler seus próprios dados
    match /users/{userId} {
      allow read, write: if request.auth != null && request.auth.uid == userId;
    }

    // Alunos - leitura para todos autenticados
    match /students/{studentId} {
      allow read: if request.auth != null;

      // Escrita apenas para colaboradores
      allow write: if request.auth != null &&
        get(/databases/{database}/documents/users/{request.auth.uid}).data.role == 'colaborador';
    }
  }
}
```

## 10.4 Restrições e Exceções

### Operações Sensíveis

- **Exclusão de alunos:** Não implementada (apenas inativação)
- **Alteração de roles:** Apenas via console Firebase (administrador)
- **Recuperação de dados:** Backup manual semanal por colaborador designado

### Exceções Temporárias

- **Professores em período de matrícula:** Podem receber permissão temporária de edição
- **Situações de emergência:** Colaborador pode criar usuário professor temporário
- **Relatórios especiais:** Professor coordenador pode ter acesso a exportações

### Auditoria e Logs

- **Ações sensíveis logadas:** Criação, edição, mudança de status
- **Acessos registrados:** Firebase Analytics registra tentativas de acesso

- **Revisão mensal:** Colaborador revisa logs de ações realizadas

## 10.5 Processo de Concessão de Acesso

### Novo Usuário Professor

1. Colaborador cria conta no Firebase Authentication
2. Sistema cria documento em **users** com role 'professor'
3. Usuário recebe credenciais temporárias via email seguro
4. Primeiro login força mudança de senha

### Novo Usuário Colaborador

1. Administrador de sistema cria conta manualmente
2. Validação presencial da identidade
3. Definição de senha segura em processo supervisionado
4. Teste de todas as funcionalidades com usuário novo

### Revogação de Acesso

1. Desativação imediata na Firebase Authentication
2. Documento em **users** marcado como inativo
3. Revisão de todas as ações realizadas pelo usuário
4. Notificação para outros colaboradores

---

## 11. ARQUITETURA E STACK TECNOLÓGICO

### 11.1 Visão Geral da Arquitetura

Frontend (Next.js + React)	Serviços (Firebase)	Banco de Dados (Firestore)
<ul style="list-style-type: none"> <li>• Interface Web</li> <li>• Componentes</li> <li>• Estado Global</li> <li>• Validações</li> </ul>	<ul style="list-style-type: none"> <li>• Autenticação</li> <li>• Security Rules</li> <li>• Analytics</li> <li>• Hosting</li> </ul>	<ul style="list-style-type: none"> <li>• NoSQL</li> <li>• Tempo Real</li> <li>• Offline</li> <li>• Backup</li> </ul>

### 11.2 Stack Frontend

#### Next.js 15 + React 19

- **Justificativa:** Framework moderno com SSR, otimizações automáticas
- **Características:** App Router, Turbopack, TypeScript nativo

- **Performance:** Code splitting automático, lazy loading

### TypeScript

- **Justificativa:** Tipagem estática reduz bugs, melhora DX
- **Configuração:** Strict mode ativado
- **Integração:** Types para Firebase SDK, React Hook Form

### Tailwind CSS 4.0

- **Justificativa:** Utility-first, design system consistente
- **Responsividade:** Mobile-first approach
- **Performance:** PurgeCSS integrado, CSS otimizado

### React Hook Form + Zod

- **Justificativa:** Performance superior, validação declarativa
- **Características:** Uncontrolled components, validação assíncrona
- **Integração:** Schema validation com TypeScript inference

## 11.3 Stack Backend e Serviços

### Firebase Authentication

- **Funcionalidades:** Email/password, session management
- **Segurança:** Token-based auth, automatic token refresh
- **Escalabilidade:** Suporte a milhões de usuários

### Cloud Firestore

- **Tipo:** NoSQL document database
- **Vantagens:** Realtime updates, offline support, automatic scaling
- **Estrutura:** Coleções `users` e `students`
- **Security Rules:** Controle granular de acesso

### Firebase Analytics

- **Propósito:** Monitoramento de uso, performance insights
- **Métricas:** Page views, user actions, error tracking
- **Privacy:** GDPR compliant, configuração de dados anônimos

## 11.4 Bibliotecas Auxiliares

### Validação e Formulários

```
{
  "react-hook-form": "^7.63.0",
  "@hookform/resolvers": "^5.2.2",
```

```
    "zod": "^4.1.11"
  }
```

### UI e Ícones

```
{
  "lucide-react": "^0.544.0",
  "tailwindcss": "^4"
}
```

### Utilidades

```
{
  "papaparse": "^5.5.3",
  "date-fns": "^4.1.0"
}
```

## 11.5 Fluxo de Dados

### Autenticação

1. Login via Firebase Auth
2. Token JWT armazenado no localStorage
3. Context API gerencia estado global do usuário
4. Verificação automática de expiração

### Operações CRUD

1. Interface disparar ação
2. Validação client-side (Zod)
3. Chamada Firebase SDK
4. Security Rules validam server-side
5. Firestore executa operação
6. Realtime listener atualiza UI

### Estados de Loading

```
const [loading, setLoading] = useState(false);
const [error, setError] = useState<string | null>(null);

try {
  setLoading(true);
  await firestoreOperation();
} catch (err) {
  setError(err.message);
} finally {
  setLoading(false);
}
```

## 11.6 Deploy e DevOps

### Desenvolvimento

- **Comando:** `npm run dev`
- **Features:** Hot reload, Turbopack bundler
- **Porta:** `http://localhost:3000`

### Build de Produção

- **Comando:** `npm run build`
- **Otimizações:** Tree shaking, minification, compression
- **Validação:** TypeScript check, ESLint

### Deploy (Vercel)

- **Processo:** Git push → Build automático → Deploy
- **Features:** Preview deployments, custom domains
- **Performance:** Edge CDN, automatic HTTPS

### Monitoramento

- **Firebase Console:** Database usage, authentication metrics
- **Vercel Analytics:** Performance insights, Web Vitals
- **Error Tracking:** `Console.error` logs, crash reports

## 11.7 Segurança

### Autenticação e Autorização

- JWT tokens com expiração automática
- RBAC implementado em Security Rules
- Session timeout configurável

### Proteção de Dados

- HTTPS obrigatório em produção
- Firestore Security Rules impedem acesso não autorizado
- Validação dupla (client + server)

### Privacy e LGPD

- Dados pessoais criptografados em trânsito
- Logs não contêm informações identificáveis
- Direito ao esquecimento implementável

## 12. CRITÉRIOS DE ACEITE

### 12.1 RF-AUTH-001: Login do Sistema

Feature: Autenticação de usuário

Como um professor ou colaborador

Quero fazer login no sistema

Para acessar as funcionalidades da plataforma

Scenario: Login bem-sucedido com credenciais válidas

Given que eu tenho uma conta válida no sistema

And estou na página de login

When eu preencho o campo email com "maria@creche.com"

And eu preencho o campo senha com "senha123"

And eu clico no botão "Entrar"

Then eu devo ser redirecionado para o dashboard

And devo ver meu nome "Maria Santos" no cabeçalho

And devo ter acesso às funcionalidades do meu perfil

Scenario: Login falha com credenciais inválidas

Given que estou na página de login

When eu preencho o campo email com "usuario@inexistente.com"

And eu preencho o campo senha com "senhaerrada"

And eu clico no botão "Entrar"

Then devo ver a mensagem "Email ou senha incorretos. Tente novamente."

And devo permanecer na página de login

And os campos devem permanecer preenchidos

Scenario: Validação de campos obrigatórios

Given que estou na página de login

When eu deixo o campo email vazio

And eu preencho o campo senha com "senha123"

And eu clico no botão "Entrar"

Then devo ver a mensagem "Email inválido"

And o botão "Entrar" deve permanecer desabilitado

And não deve haver tentativa de autenticação

### 12.2 RF-STUD-001: Cadastro de Aluno

Feature: Cadastro de novo aluno

Como um colaborador

Quero cadastrar um novo aluno

Para incluí-lo no sistema da creche

Scenario: Cadastro bem-sucedido com todos os dados obrigatórios

Given que estou logado como colaborador

And estou na aba "Cadastrar"



When eu preencho "Nome do Aluno" com "João da Silva"  
 And eu preencho "Data de Nascimento" com "15/03/2020"  
 And eu preencho "Nome do Responsável" com "Ana Silva"  
 And eu preencho "Telefone" com "(11) 99999-9999"  
 And eu preencho "Email" com "ana.silva@email.com"  
 And eu preencho "Endereço" com "Rua das Flores, 123, São Paulo"  
 And eu preencho "Contato de Emergência" com "Carlos Silva"  
 And eu preencho "Telefone de Emergência" com "(11) 88888-8888"  
 And eu preencho "Data de Matrícula" com "01/02/2025"  
 And eu seleciono "Status" como "Ativo"  
 And eu clico em "Cadastrar Aluno"  
 Then devo ver a mensagem " Aluno cadastrado com sucesso!"  
 And o formulário deve ser limpo para novo cadastro  
 And o aluno deve aparecer na lista com idade calculada "4 anos"

Scenario: Falha no cadastro por campos obrigatórios vazios  
 Given que estou logado como colaborador  
 And estou na aba "Cadastrar"  
 When eu deixo o campo "Nome do Aluno" vazio  
 And eu preencho os demais campos obrigatórios  
 And eu clico em "Cadastrar Aluno"  
 Then devo ver a mensagem "Nome deve ter pelo menos 2 caracteres"  
 And o campo "Nome do Aluno" deve estar destacado em vermelho  
 And o aluno não deve ser salvo  
 And devo permanecer no formulário

Scenario: Validação de formato de email  
 Given que estou logado como colaborador  
 And estou preenchendo o cadastro de aluno  
 When eu preencho "Email do Responsável" com "email-inválido"  
 And eu tento sair do campo  
 Then devo ver a mensagem "Email inválido"  
 And o campo deve estar destacado em vermelho  
 And o botão "Cadastrar" deve estar desabilitado

### 12.3 RF-STUD-002: Listagem de Alunos

Feature: Visualizar lista de alunos  
 Como um professor ou colaborador  
 Quero ver a lista de alunos cadastrados  
 Para consultar suas informações

Scenario: Carregar lista completa de alunos  
 Given que estou logado no sistema  
 And existem 15 alunos cadastrados no sistema  
 When eu acesso a aba "Alunos"

Then devo ver uma lista com 15 alunos  
And cada aluno deve mostrar: nome, idade, responsável, telefone, data de matrícula, status  
And a lista deve estar ordenada alfabeticamente por nome  
And devo ver o contador "Lista de Alunos (15)"

Scenario: Lista vazia quando não há alunos  
Given que estou logado no sistema  
And não existem alunos cadastrados  
When eu acesso a aba "Alunos"  
Then devo ver a mensagem "Nenhum aluno cadastrado ainda."  
And devo ver o ícone de usuários  
And o contador deve mostrar "Lista de Alunos (0)"

Scenario: Exibição de botão editar apenas para colaboradores  
Given que estou logado como professor  
And estou visualizando a lista de alunos  
Then não devo ver botões "Editar" nos alunos  
And devo conseguir ver todas as informações dos alunos

Given que estou logado como colaborador  
And estou visualizando a lista de alunos  
Then devo ver botões "Editar" em todos os alunos  
And devo conseguir clicar nos botões de edição

## 12.4 RF-STUD-003: Busca e Filtros

Feature: Buscar e filtrar alunos  
Como um usuário do sistema  
Quero buscar alunos por nome e aplicar filtros  
Para encontrar rapidamente informações específicas

Scenario: Busca por nome em tempo real  
Given que estou na lista de alunos  
And existem alunos com nomes "João Silva", "Maria João", "Pedro Santos"  
When eu digito "João" no campo de busca  
Then devo ver apenas os alunos "João Silva" e "Maria João"  
And a busca deve acontecer instantaneamente sem clicar em botão  
And o contador deve mostrar "Lista de Alunos (2)"

Scenario: Filtro por faixa etária  
Given que estou na lista de alunos  
And existem alunos de diferentes idades  
When eu seleciono o filtro "2-3 anos"  
Then devo ver apenas alunos com idade entre 2 e 3 anos  
And outros alunos devem ser ocultados  
And posso combinar com busca por nome

Scenario: Combinação de múltiplos filtros  
Given que estou na lista de alunos  
When eu digito "Maria" no campo de busca  
And eu seleciono filtro "4-5 anos"  
And eu seleciono status "Ativo"  
Then devo ver apenas alunos que atendem TODOS os critérios  
And devo poder limpar todos os filtros de uma vez

Scenario: Nenhum resultado encontrado  
Given que estou na lista de alunos  
When eu busco por "XYZ123" (nome inexistente)  
Then devo ver a mensagem "Nenhum aluno encontrado"  
And devo ver "Tente ajustar os filtros de busca."  
And devo poder limpar a busca facilmente

## 12.5 RF-REP-002: Exportação CSV

Feature: Exportar relatórios em CSV  
Como um colaborador  
Quero exportar dados de alunos em CSV  
Para análises externas e backup

Scenario: Exportar todos os alunos  
Given que estou logado como colaborador  
And estou na aba "Relatórios"  
And existem 10 alunos cadastrados  
When eu clico no botão "Todos os Alunos" na seção de exportação  
Then deve iniciar o download de um arquivo CSV  
And o arquivo deve se chamar "relatorio-completo-AAAAMDD-HHMMSS.csv"  
And o arquivo deve conter 10 linhas de dados + 1 linha de cabeçalho  
And deve incluir todas as colunas: Nome, Idade, Data de Nascimento, etc.

Scenario: Exportar apenas alunos ativos  
Given que estou logado como colaborador  
And existem 8 alunos ativos e 2 inativos  
When eu clico no botão "Alunos Ativos"  
Then o arquivo CSV deve conter apenas 8 linhas de dados  
And todos os alunos no arquivo devem ter status "Ativo"

Scenario: Exportar por faixa etária  
Given que estou na seção de relatórios  
And existem 3 alunos na faixa "2-3 anos"  
When eu clico no botão da faixa "2-3 anos (3)"  
Then deve baixar arquivo "alunos-2-3-anos-AAAAMDD-HHMMSS.csv"  
And deve conter apenas os 3 alunos da faixa etária

Scenario: Botão desabilitado quando não há dados  
Given que estou na seção de relatórios  
And não existem alunos inativos  
When eu visualizo o botão "Alunos Inativos"  
Then o botão deve estar desabilitado  
And deve mostrar "Alunos Inativos (0)"  
And não deve ser possível clicar

Scenario: Professor não pode exportar  
Given que estou logado como professor  
When eu acesso a aba "Relatórios"  
Then devo ver as estatísticas normalmente  
But não devo ver os botões de exportação CSV  
And devo ver apenas informações de consulta

---

## CONCLUSÃO

Esta documentação técnica apresenta de forma completa e estruturada o **Sistema de Gestão de Matrículas - Dashboard Creche Estrela do Oriente**, atendendo a todos os requisitos especificados para documentação profissional.

O sistema desenvolvido em **Next.js 15 + Firebase** oferece uma solução moderna, escalável e segura para a gestão digitalizada de matrículas em creches, eliminando processos manuais e centralizando informações críticas dos alunos.

### Principais Conquistas:

**Digitalização Completa** - Fim dos processos manuais e planilhas dispersas  
**Interface Moderna** - Design responsivo e acessível para todos os dispositivos  
**Segurança Robusta** - Controle de acesso baseado em roles com Firebase Auth  
**Relatórios Automatizados** - Geração instantânea de estatísticas e exportações CSV  
**Performance Otimizada** - Build sem erros, componentes otimizados, Web Vitals excelentes

### Impacto Esperado:

- **Redução de 70%** no tempo gasto em tarefas administrativas
- **Acesso instantâneo** às informações de qualquer dispositivo
- **Eliminação de riscos** de perda de documentos físicos
- **Base sólida** para crescimento e novas funcionalidades

O sistema está **pronto para produção** e pode ser implementado imediatamente, oferecendo valor significativo tanto para os usuários quanto para a organização educacional.

---

**Documento aprovado e validado**

**Data:** 07 de outubro de 2025

**Versão:** 1.0.0

**Próxima revisão:** Janeiro de 2026