

RELATÓRIO DE DESENVOLVIMENTO DO SISTEMA

Sistema de Gestão do Programa Escola de Tempo Integral — Instituto IDS

RESUMO EXECUTIVO: Este documento apresenta a especificação técnica completa, arquitetura de software, modelo de dados relacional, controle de acesso granular (RBAC), módulos funcionais e a pilha de tecnologias utilizadas no desenvolvimento do **Sistema de Gestão do Programa Escola de Tempo Integral**.

REGIME DE CONTRATAÇÃO: Todos os colaboradores e educadores são obrigatoriamente cadastrados sob o regime **Pessoa Física / CLT** (Consolidação das Leis do Trabalho) com retenções legais de INSS, IRRF e FGTS.

1. Visão Geral do Sistema

O **Sistema de Gestão do Programa Escola de Tempo Integral** foi concebido para centralizar a administração, operação pedagógica, acompanhamento de recursos humanos no regime **Pessoa Física / CLT**, controle orçamentário-financeiro e avaliação de qualidade das unidades escolares atendidas pelo **Instituto IDS**.

A plataforma conecta em tempo real gestores escolares, coordenadores, operadores gerais e administradores, proporcionando visibilidade consolidada do desempenho operacional e pedagógico através de dashboards intuitivos e relatórios analíticos de Monitoramento & Avaliação (M&A;).

2. Arquitetura da Solução e Pilha Tecnológica

A solução adota uma arquitetura descentralizada com desacoplamento entre a camada visual (Single Page Application - SPA) e o backend de microsserviços/REST API com persistência relacional robusta.

A) Frontend (SPA)

- Framework Core:** React 18 com bundler Vite (alto desempenho de compilação e carregamento).
- Roteamento:** React Router DOM v6 com suporte a rotas protegidas e suporte a subdomínios dinâmicos para formulários públicos.
- Gerenciamento de Estado Assíncrono:** TanStack React Query v5 (cache eficiente e revalidação de dados).
- Estilização e Componentes:** Tailwind CSS 3.4, primitivos acessíveis Radix UI e biblioteca de ícones Lucide React.
- Visualização de Dados & PDFs:** Recharts para dashboards analíticos; jsPDF e html2canvas para relatórios exportáveis.

B) Backend & Banco de Dados

- Servidor REST API:** Python 3 + Flask com suporte a CORS e manipulação de erros customizados.
- Camada ORM & Banco:** SQLAlchemy conectado a banco de dados relacional PostgreSQL.
- Segurança & Autenticação:** Tokens JWT (JSON Web Token) e criptografia de senhas Werkzeug Security.
- Modelagem BaaS:** Especificações de entidades declarativas em esquemas JSONC (Base44).

3. Controle de Acesso e Perfis de Usuários (RBAC)

O sistema implementa um modelo rígido de Controle de Acesso Baseado em Funções (Role-Based Access Control), assegurando a proteção e a privacidade dos dados conforme o perfil funcional de cada usuário:

Perfil (Role)	Escopo de Acesso e Permissões
---------------	-------------------------------

dev / admin	Acesso irrestrito a todas as visões, relatórios, cadastros, aprovação de solicitações de acesso e logs de auditoria.
coordinator	Acompanhamento de escolas, alunos, professores, modalidades pedagógicas, indicadores M&A; e semáforo operacional.
general_operator	Operação de Recursos Humanos (CLT), Folha de Pagamento, Gestão Financeira, Materiais e cadastros operacionais.
school_manager	Acesso focado na aplicação e visualização de formulários de pesquisa da sua respectiva unidade escolar.
Público (Subdomínio)	Acesso sem login direto no subdomínio de formulários para preenchimento de pesquisas pela comunidade escolar.

4. Módulos Funcionais do Sistema

- **Dashboard Executivo:** KPIs consolidados de escolas, alunos, professores, modalidades ativas e execução orçamentária acumulada.
- **Gestão Pedagógica:** Módulos para cadastro e enturmação de Alunos, alocação de Professores, gestão de Escolas e parametrização de Modalidades/Oficinas de tempo integral.
- **Recursos Humanos e Folha (Pessoa Física / CLT):** Cadastro exclusivo sob o regime **Pessoa Física / CLT** (com identificador obrigatório CPF, carteira de trabalho, dependentes e conta salário), upload de documentos comprobatórios e emissão de Folha de Pagamento mensal com cálculo progressivo automático de **INSS, IRRF e FGTS**.
- **Gestão Financeira & Materiais:** Controle de entradas e saídas por Centro de Custo e Rubrica, acompanhamento de insumos pedagógicos e módulo de Prestação de Contas.
- **Subsistema de Monitoramento & Qualidade (M&A;):** Formulário público/privado de pesquisa de satisfação, cálculo automatizado dos índices **ISGE** e **NPS**, Semáforo Operacional (Verde/Amarelo/Vermelho), Registro de Ocorrências/Alertas e Relatórios Periódicos M&A.;

5. Modelo de Dados e Principais Entidades

A base de dados é estruturada em tabelas relacionais no PostgreSQL, mapeadas pelo SQLAlchemy ORM e espelhadas em esquemas declarativos:

Entidade	Campos Principais e Descrição
User	id, name, email, password_hash, role (dev, admin, coordinator, general_operator, school_manager)
AccessRequest	id, name, email, password_hash, status (pending, approved, rejected), created_at
School	id, name, gestor_responsavel, quantidade_alunos, endereco
Employee	id, name, cpf (obrigatório - Pessoa Física), rg, birth_date, phone, email, bank, agency, account, pix, contract_type (CLT/Pessoa Física), contract_value, status
Payroll	id, competencia, status, folha_number, created_date, items (JSON com INSS, IRRF e FGTS)
CostCenter	id, name, type, schools (JSON), codigo, rubrica, categoria
SchoolEvaluation	id, school_id, school_name, evaluation_date, isge, nps, traffic_light, notes
Alert	id, title, description, alert_date, school_id, status (Pendente, Em Resolução, Resolvido)
AuditLog	id, user_id, user_name, action, details, created_date

6. Destaques de Engenharia e Qualidade

- **Desacoplamento de Utilitários:** Cálculos de tributação CLT (INSS, IRRF, FGTS) isolados e validados no módulo payrollCalculations.js.
- **Tratamento de Subdomínio:** Detecção automática do hostname no frontend para servir o formulário de pesquisa sem exigência de credenciais.
- **Resiliência com Cache:** React Query gerenciando o ciclo de vida e estado das requisições assíncronas.
- **Compilação e Build Limpo:** Projeto testado e validado via bundler Vite com zero erros de compilação.

7. Recomendações Técnicas e Roadmap de Evolução

1. **Testes Automatizados:** Implementação de testes E2E com Playwright e suítes unitárias em PyTest no backend.
2. **Observabilidade:** Rastreamento centralizado de logs e exceções em tempo de execução via Sentry.
3. **Paginação Server-Side:** Expansão da paginação na API para listas de grande escala (Alunos e Respostas).
4. **Segurança Contínua:** Rotação automatizada de tokens JWT e backup programado do PostgreSQL.