

# Documentação Final do Projeto

## Sistema de Gestão de Projetos Integradores (SGPI)

**Desenvolvedores:**

Diego Marques da Costa  
Kaique Caitano dos Santos  
Luiz Carlos Gimenes Fernandes de Sousa  
Nicole Lisboa de Carvalho  
Vinicius de Moraes Crepaldi

**Instituição:** FATEC Zona Leste

**Curso:** Desenvolvimento de Software Multiplataforma

# Sumário

<b>1. ARQUITETURA E IMPLEMENTAÇÃO TÉCNICA .....</b>	<b>5</b>
1.1 Estrutura do Projeto .....	5
1.1.1 Primeira Etapa: Modelo MVC .....	5
1.1.2 Segunda Etapa: Helpers e Middlewares .....	5
1.1.3 Terceira Etapa: Separação de Conteúdo JavaScript Frontend.....	5
1.1.4 Quarta Etapa: Refatoração JavaScript Frontend .....	5
1.1.5 Quinta Etapa: Refatoração CSS Frontend .....	5
1.2 Fases de Implementação .....	5
FASE 1 - Padrão de Respostas e Tratamento Global de Erros .....	5
FASE 2 - Conexão Oracle e Desempenho .....	6
FASE 3 - Autenticação, Autorização e Segurança .....	6
FASE 4 - Validação e Consistência de Dados .....	6
FASE 5 - Padronização de Rotas e Paginação/Filtragem .....	6
FASE 6 - Upload e Imagens de Perfil (Consolidação) .....	7
FASE 7 - Domínio SGPI e Regras de Negócio .....	7
FASE 8 - Notificações e Tarefas Agendadas .....	7
FASE 9 - Observabilidade e Saúde do Sistema .....	7
FASE 10 - Migrações, Seeds e Integridade .....	7
1.3 Cronograma de Execução .....	8
<b>2. FRONT-END .....</b>	<b>8</b>
2.1 Estrutura e Navegação .....	8
2.1.1 Aplicação de Navegabilidade e Criação de Footer .....	8
2.1.2 Reestruturação do Menu de Perfil .....	8
2.1.3 Atualização da Dinâmica Interna da Página 'Perfil' .....	8
2.1.4 Padronização de Estilização .....	8
2.2 Funcionalidades de Cadastro e Autenticação .....	8
2.2.1 Adição da Dinâmica de Tipo de Conta .....	8
2.2.2 Controle de Palavras-Chave .....	9
2.3 Gestão Acadêmica .....	9
2.3.1 Painel de Semestres para Administrador .....	9
2.3.2 Gerenciamento de Disciplinas .....	9
2.3.3 Painel de Ativação/Desativação do Período de Matrículas .....	9
2.4 Gestão de Projetos e Grupos .....	9

2.4.1	CRUD da Tela de Grupos .....	9
2.4.2	CRUD dos Projetos.....	9
2.4.3	Ajustes na Lista de Projetos do Coordenador .....	9
2.5	Sistema de Atividades.....	9
2.5.1	CRUD das Atividades dos Professores .....	9
2.5.2	Lista de Tarefas do Aluno.....	9
2.5.3	Página de Entrega de Arquivos .....	10
2.5.4	Página de Lista de Avaliações dos Alunos.....	10
2.5.5	Página de Lista dos Pedidos de Reconsideração .....	10
2.6	Sistema de Notificações .....	10
2.6.1	E-mails Funcionais.....	10
2.7	Relatórios.....	10
2.7.1	Ajuste na Tela de Gerar Relatórios.....	10
2.8	Recursos de Sistema .....	10
2.8.1	Adição do Plugin de Acessibilidade .....	10
2.8.2	Criação de Termos de Uso .....	10
2.8.3	Criação da Página "Sobre" .....	10
2.8.4	Política de Privacidade e Termos de Uso .....	11
2.9	Otimizações .....	11
2.9.1	Otimização da Imagem de Perfil.....	11
2.9.2	Adequação de Função Utilitária .....	11
2.9.3	Padronização de Confirmações .....	11
<b>3.</b>	<b>BACK-END .....</b>	<b>11</b>
3.1	Migração e Infraestrutura .....	11
3.1.1	Migração de SGBD .....	11
3.1.2	Migração Parcial do Cadastro .....	11
3.2	Sistema de Autenticação .....	11
3.2.1	Backend da Função de Palavras-Chave.....	11
3.2.2	Dinâmica das Palavras-Chave .....	11
3.3	Gestão Acadêmica .....	12
3.3.1	Rotas de Disciplinas.....	12
3.3.2	Rotas do Painel de Matrículas.....	12
3.4	Gestão de Projetos e Grupos.....	12
3.4.1	Adequação das Rotas de Grupos .....	12
3.4.2	Adequação das Rotas de Projetos.....	12
3.4.3	Rotas da Lista de Projetos do Coordenador .....	12

3.5 Sistema de Atividades.....	12
3.5.1 CRUD das Rotas de Atividades .....	12
3.5.2 Rotas Refatoradas de Atividades dos Alunos .....	12
3.5.3 Rotas Refatoradas de Reconsideração.....	12
3.6 Relatórios.....	12
3.6.1 Rotas de Geração de Relatórios do Coordenador .....	12
3.7 Gerenciamento de Perfil .....	13
3.7.1 Imagem de Perfil .....	13
3.8 Otimizações .....	13
3.8.1 Reformulação das Respostas de Array do MySQL .....	13
<b>4. BANCO DE DADOS.....</b>	<b>13</b>
4.1 Modelagem e Estrutura .....	13
4.1.1 Modelagem do Banco de Dados.....	13
4.1.2 Script do Banco de Dados.....	13
4.1.3 Script do Banco e Novo SGBD.....	13
4.2 Validação e Testes.....	13
4.2.1 Teste de Mesa .....	13
4.2.2 Teste de Mesa Finalizado.....	13
4.2.3 Teste de Selects, Inserts, Join e Procedures .....	13
4.3 Otimizações .....	14
4.3.1 Melhorias na Lógica dos Dados.....	14
4.3.2 Criação de Índices .....	14
4.4 Estruturas Específicas .....	14
4.4.1 Alteração da Tabela de Grupos .....	14
4.5 Implementação Final.....	14
4.5.1 Migração do Banco Completa.....	14
<b>5. PLANO DE TESTES FUNCIONAIS E DE USABILIDADE .....</b>	<b>14</b>
5.1 Testador 1 .....	14
5.2 Testador 2 .....	14
5.3 Testador 3 .....	14
5.4 Testador 4 .....	14
<b>6. CONSIDERAÇÕES FINAIS .....</b>	<b>15</b>
6.1 Tecnologias Utilizadas .....	15
6.2 Padrões e Boas Práticas .....	15

# 1. ARQUITETURA E IMPLEMENTAÇÃO TÉCNICA

## 1.1 Estrutura do Projeto

### 1.1.1 Primeira Etapa: Modelo MVC

Criação do modelo MVC para organização dos arquivos cruciais:

- **Routes:** Regras de negócio
- **Views:** Interfaces de usuário
- **JavaScript:** Módulos

Implementação de RBAC (Role-Based Access Control) por papel (Aluno, Professor, Professor\_Orientador, Coordenador) via middleware nas rotas.

### 1.1.2 Segunda Etapa: Helpers e Middlewares

Criação de pastas para organização do backend:

- **Helpers:** Funções auxiliares
- **Middlewares:** Funções de processamento de requisições intermediárias

### 1.1.3 Terceira Etapa: Separação de Conteúdo JavaScript Frontend

Organização do código frontend em pastas específicas:

- **front\_end:** Manipulação visual e eventos
- **services:** Comunicação com backend
- **utils:** Funções auxiliares puras

### 1.1.4 Quarta Etapa: Refatoração JavaScript Frontend

Refatoração dos arquivos JavaScript do frontend para separação das funções específicas de cada módulo.

### 1.1.5 Quinta Etapa: Refatoração CSS Frontend

Refatoração dos arquivos CSS para:

- Separação de código duplicado
- Organização estrutural
- Melhoria da escalabilidade do design do sistema

## 1.2 Fases de Implementação

### FASE 1 - Padrão de Respostas e Tratamento Global de Erros

1. Definição de contrato de resposta padronizado: {success, data, message, errorCode}
2. Implementação de middleware global de erros
3. Mapeamento consistente de status HTTP (400, 401, 403, 404, 409, 422, 500)
4. Padronização de mensagens em todas as rotas críticas (login, cadastro, perfil, redefinição de senha, projetos)

## **Critérios de Aceite:**

- Requisições inválidas retornam 400 com errorCode preciso
- Respostas seguem padrão estabelecido

## **FASE 2 - Conexão Oracle e Desempenho**

1. Implementação de pool de conexões (createPool) e reuso por requisição
2. Fechamento adequado do pool no shutdown
3. Revisão de consultas comuns e criação de índices complementares
4. Implementação de índice funcional UPPER(email) para login case-insensitive

## **Critérios de Aceite:**

- Latência média reduzida em endpoints de leitura
- Consumo de conexões estável sob carga leve

## **FASE 3 - Autenticação, Autorização e Segurança**

1. Implementação de CORS restrito (origens específicas)
2. Configuração do Helmet para headers de segurança
3. Rate limiting em endpoints sensíveis (login, redefinição)
4. Garantia de que campos sensíveis (senha, tokens) nunca sejam expostos nas respostas
5. Sanitização de logs

## **Critérios de Aceite:**

- Rotas protegidas por papel
- Tentativas acima do limite bloqueadas
- Login e redefinição auditados

## **FASE 4 - Validação e Consistência de Dados**

1. Validação de entrada com schema (Joi/Zod/Yup) para cadastro, login, perfil, criação de atividades/projetos/entregas
2. Normalização de binds nomeados nas queries
3. Eliminação de SELECT \* e retorno apenas de colunas necessárias
4. Validação de conteúdo em uploads (assinatura/magic number), tamanho e dimensões
5. Remoção de metadados EXIF

## **Critérios de Aceite:**

- Requisições inválidas retornam 400 com errorCode preciso
- Uploads não-imagem são rejeitados

## **FASE 5 - Padronização de Rotas e Paginação/Filtragem**

1. Introdução de prefixo de versão: /api/v1
2. Manutenção de rotas antigas por um ciclo de depreciação
3. Padronização de coleções com:
  - Paginação (page, pageSize)
  - Ordenação (sort, order)
  - Filtros consistentes
4. Aplicação em: usuários, grupos, projetos, atividades, entregas, avaliações, reconsiderações, notificações

### **Critérios de Aceite:**

- Endpoints de listagem com paginação funcional
- Parâmetros documentados
- Rotas antigas continuam operando

### **FASE 6 - Upload e Imagens de Perfil (Consolidação)**

1. Confirmação de uso exclusivo do BLOB em Usuarios.foto
2. Remoção de dependência de /imagens/perfil do disco
3. Implementação de validação rigorosa de uploads

### **Critérios de Aceite:**

- GET da foto retorna ETag
- Revalidação funciona
- PUT de foto reflete nova ETag

### **FASE 7 - Domínio SGPI e Regras de Negócio**

1. Implementação de transições de status de projetos com histórico
2. Conexão com trigger/Histórico no banco de dados
3. Justificativas aplicacionais

### **Critérios de Aceite:**

- Fluxos cobertos por testes de integração
- Violação de regra retorna 403/409 com errorCode

### **FASE 8 - Notificações e Tarefas Agendadas**

1. Orquestração de notificações de prazos com job scheduler no app (cron)
2. Chamada de procedures do banco com logs e retries
3. Alternativa: uso de DBMS\_SCHEDULER com observabilidade do lado do app
4. Consulta e registro de execuções

### **Critérios de Aceite:**

- Logs de execução dos jobs
- Notificações geradas conforme regras de negócio
- Mecanismos de retry e alerta de falhas

### **FASE 9 - Observabilidade e Saúde do Sistema**

1. Implementação de endpoint /health (básico: up; avançado: ping no banco)
2. Logging estruturado (JSON) com correlationId por requisição
3. Níveis de log (info/warn/error)
4. Métricas básicas (contagem de requisições, latência, erros por rota)
5. Opção para Prometheus no futuro
6. Melhoria de cache: ETag/Last-Modified com suporte a 304 Not Modified
7. Fallback seguro no frontend mantido

### **Critérios de Aceite:**

- Sistema de monitoramento operacional
- Logs estruturados e rastreáveis

### **FASE 10 - Migrações, Seeds e Integridade**

1. Versionamento de scripts SQL (migrations)

2. Seeds mínimos para ambientes de dev/test
3. Setup do banco reproduzível do zero
4. Possibilidade de rollback
5. Garantia de integridade

### **Critérios de Aceite:**

- Setup do banco reproduzível
- Rollback possível
- Integridade garantida

## **1.3 Cronograma de Execução**

**Semana 1:** Fase 1 e 2 (Erros/Padrões + Pool Oracle)

**Semana 2:** Fase 3 e 4 (JWT/RBAC + Validação)

**Semana 3:** Fase 5 e 6 (API v1 + Paginação + Imagens Consolidadas)

**Semana 4:** Fase 7 e 8 (Regras de Negócio + Jobs de Notificações)

**Semana 5:** Fase 9 e 10 (Observabilidade + Documentação)

**Semana 6:** Fase 11, 12 e 13 (Migrations, Testes, Deploy/Hardening)

# **2. FRONT-END**

## **2.1 Estrutura e Navegação**

### **2.1.1 Aplicação de Navegabilidade e Criação de Footer**

- Implementação de menu horizontal principal
- Menu dropdown adaptativo para versões mobile
- Criação de footer com links úteis do sistema
- Navegação dinâmica e intuitiva

### **2.1.2 Reestruturação do Menu de Perfil**

- Remoção da estrutura do ícone de perfil para a barra horizontal
- Criação do ícone de notificações internas fixado ao lado do ícone de perfil

### **2.1.3 Atualização da Dinâmica Interna da Página 'Perfil'**

- Implementação de barra lateral com opções relevantes para cada tipo de usuário
- Integração com backend via API para carregamento de conteúdo

### **2.1.4 Padronização de Estilização**

- Modernização das telas de login, cadastro, esqueci a senha e redefinir senha
- Atualização do modal de validação de token do cadastro
- Novo modelo visual mais moderno

## **2.2 Funcionalidades de Cadastro e Autenticação**

### **2.2.1 Adição da Dinâmica de Tipo de Conta**

- Troca do semestre por seleção de matérias



- Dinâmica de semestre implementada no banco de dados através dos relacionamentos das matérias selecionadas

### **2.2.2 Controle de Palavras-Chave**

- Criação do painel de palavras-chave para o tipo de conta coordenador
- Geração de palavras de 8 dígitos alfanuméricos
- Sistema de desativação de palavras-chave com número limitado de usos
- Prevenção de uso excessivo e aumento da segurança

## **2.3 Gestão Acadêmica**

### **2.3.1 Painel de Semestres para Administrador**

- Funcionalidade para adicionar novos semestres
- Transformação de semestres anteriores em histórico

### **2.3.2 Gerenciamento de Disciplinas**

- Desenvolvimento da aba de gerenciar disciplinas para perfis de alunos
- Implementação para professores e professores orientadores

### **2.3.3 Painel de Ativação/Desativação do Período de Matrículas**

- Controle de vínculo das disciplinas para alunos e professores
- Funcionalidade para coordenador ativar/desativar conforme período letivo

## **2.4 Gestão de Projetos e Grupos**

### **2.4.1 CRUD da Tela de Grupos**

- Implementação completa do CRUD de grupos
- Adição de função auxiliar de confirmação de ações

### **2.4.2 CRUD dos Projetos**

- Implementação completa do CRUD de projetos
- Função auxiliar de confirmação de ações

### **2.4.3 Ajustes na Lista de Projetos do Coordenador**

- Otimização da interface de listagem de projetos

## **2.5 Sistema de Atividades**

### **2.5.1 CRUD das Atividades dos Professores**

- Create, Update, Read e Delete totalmente funcionais no front e back
- Integração completa com backend

### **2.5.2 Lista de Tarefas do Aluno**

- Filtro funcional da lista de atividades
- Dinâmica de atividade atrasada que desabilita o envio

### **2.5.3 Página de Entrega de Arquivos**

- Entrega e atualização funcionais com filtragem
- Sistema que previne múltiplas entregas quando um membro do grupo já entregou
- Adição de confirmação nas entregas através de dialogConfirm.js

### **2.5.4 Página de Lista de Avaliações dos Alunos**

- Visualização de notas das atividades corrigidas
- Possibilidade de pedido de reconsideração de notas em prazo de 15 dias

### **2.5.5 Página de Lista dos Pedidos de Reconsideração**

- Interface para professores/orientadores visualizarem pedidos de reconsideração
- Sistema baseado nas atividades devolvidas para revisão de notas

## **2.6 Sistema de Notificações**

### **2.6.1 E-mails Funcionais**

- E-mails de avisos de atividades criadas
- E-mails de atualizações de atividades

## **2.7 Relatórios**

### **2.7.1 Ajuste na Tela de Gerar Relatórios**

- Otimização da interface de geração de relatórios

## **2.8 Recursos de Sistema**

### **2.8.1 Adição do Plugin de Acessibilidade**

- Implementação do UserWay Accessibility Widget
- Substituição do modo de contraste manual
- Adição em todas as páginas do sistema
- Recursos adicionais de acessibilidade

### **2.8.2 Criação de Termos de Uso**

- Desenvolvimento de página de termos de uso
- Versão genérica para posterior adequação às diretrizes institucionais

### **2.8.3 Criação da Página "Sobre"**

- Informações sobre o surgimento da ideia
- Benefícios para a FATEC Zona Leste e curso de DSM
- Divulgação do sistema para futuras implementações em outras FATECs

#### **2.8.4 Política de Privacidade e Termos de Uso**

- Desenvolvimento dos documentos com conteúdo elaborado pela equipe
- Implementação das páginas no sistema

### **2.9 Otimizações**

#### **2.9.1 Otimização da Imagem de Perfil**

- Implementação de compressor de imagem
- Cache usando storage para persistência dos dados da foto de perfil
- Otimização do envio de fotos pesadas
- Redução de delay e tempo de transferência para o backend

#### **2.9.2 Adequação de Função Utilitária**

- Aplicação de ES (import/export) no arquivo fetchHelper.js
- Padronização do código de produção
- Eliminação de inconsistências no chamado do JavaScript no HTML

#### **2.9.3 Padronização de Confirmações**

- Criação de arquivo utils confirmDialog.js
- Centralização de ações de confirmação no sistema

## **3. BACK-END**

### **3.1 Migração e Infraestrutura**

#### **3.1.1 Migração de SGBD**

- Migração do backend de Oracle para MySQL
- Motivação: ausência de acesso à licença do Oracle

#### **3.1.2 Migração Parcial do Cadastro**

- Migração do cadastro de aluno completa
- Implementação pendente da dinâmica de palavras-chave para professores

### **3.2 Sistema de Autenticação**

#### **3.2.1 Backend da Função de Palavras-Chave**

- Criação e armazenamento 100% funcional
- Testes de acesso validados

#### **3.2.2 Dinâmica das Palavras-Chave**

- Validação 100% da dinâmica de cadastro
- Implementação para aluno, professor e professor orientador

## **3.3 Gestão Acadêmica**

### **3.3.1 Rotas de Disciplinas**

- Implementação de rotas para vínculo de professores e orientadores
- Integração com front-end
- Rotas para atualização de disciplinas com aluno\_oferta

### **3.3.2 Rotas do Painel de Matrículas**

- Criação das rotas para controle do período de matrículas

## **3.4 Gestão de Projetos e Grupos**

### **3.4.1 Adequação das Rotas de Grupos**

- Criação de funções auxiliares para formatação de data e hora
- Implementação de filtragem de dados

### **3.4.2 Adequação das Rotas de Projetos**

- Adaptação ao novo banco de dados
- Integração com front-end

### **3.4.3 Rotas da Lista de Projetos do Coordenador**

- Ajustes nas rotas de listagem de projetos

## **3.5 Sistema de Atividades**

### **3.5.1 CRUD das Rotas de Atividades**

- Centralização da lógica em arquivo único para otimização de manutenção
- Separação por rotas /professor e /professor\_orientador
- Função de notificação por e-mail para atividades criadas e atualizadas

### **3.5.2 Rotas Refatoradas de Atividades dos Alunos**

- Unificação em arquivo único dos gets de listagem de tarefas
- CRUD de entrega e atualização de entregas
- Implementação de função de autenticação para validar ID do aluno

### **3.5.3 Rotas Refatoradas de Reconsideração**

- Migração para arquivos de centralização da lógica de fluxo de atividades
- Integração em atividadesProfessorRoutes e alunoAtividadesRoutes
- Implementação de envio de e-mails com atualizações

## **3.6 Relatórios**

### **3.6.1 Rotas de Geração de Relatórios do Coordenador**

- Ajustes nas rotas de relatórios

## **3.7 Gerenciamento de Perfil**

### **3.7.1 Imagem de Perfil**

- Descompactação dos dados da foto de perfil com shard
- Envio para banco de dados via rota de perfil

## **3.8 Otimizações**

### **3.8.1 Reformulação das Respostas de Array do MySQL**

- Criação da função utilitária extractRows no arquivo sqlUtils.js
- Adequação da devolutiva do banco de dados MySQL em JSON padrão
- Eliminação da necessidade de refatoração massiva do código

# **4. BANCO DE DADOS**

## **4.1 Modelagem e Estrutura**

### **4.1.1 Modelagem do Banco de Dados**

- Adição de tabelas associativas
- Refatoração do script inicial

### **4.1.2 Script do Banco de Dados**

- Modelo de script desenvolvido
- Preparação para validação e implantação

### **4.1.3 Script do Banco e Novo SGBD**

- Refatoração do script para adequação ao MySQL
- Adição de tabelas de histórico para consultas ao longo do tempo

## **4.2 Validação e Testes**

### **4.2.1 Teste de Mesa**

- Realização de testes com 8 cenários padrões
- Validação do script antes da implementação em produção
- Garantia de estabilidade e robustez do script

### **4.2.2 Teste de Mesa Finalizado**

- Relatório de teste de mesa realizado em laboratório
- Validação de cenários para garantir estabilidade

### **4.2.3 Teste de Selects, Inserts, Join e Procedures**

- Testes dos selects do backend refatorado no SQLOnline
- Validação funcional antes da implementação real

## 4.3 Otimizações

### 4.3.1 Melhorias na Lógica dos Dados

- Remoção de atributos desnecessários
- Criação de novos atributos para nível adequado de detalhamento

### 4.3.2 Criação de Índices

- Implementação de índices para otimizar consultas
- Melhoria de performance e desempenho do banco de dados

## 4.4 Estruturas Específicas

### 4.4.1 Alteração da Tabela de Grupos

- Adição das colunas orientador\_id e origem\_grupo\_id
- Identificação do orientador que criou os grupos
- Suporte à migração de grupos para novo semestre letivo

## 4.5 Implementação Final

### 4.5.1 Migração do Banco Completa

- Refinamento finalizado
- Script implementado
- Conexão com backend estabelecida
- Banco de dados 100% operacional

# 5. PLANO DE TESTES FUNCIONAIS E DE USABILIDADE

## 5.1 Testador 1

- **Task 07:** Chaves de Acesso
- **Task 08:** Controle de Semestres
- **Task 09:** Controle de Contas

## 5.2 Testador 2

- **Task 02:** Perfil (Visualização)
- **Task 03:** Perfil (Edição)

## 5.3 Testador 3

- **Task 04:** Disciplinas
- **Task 06:** Cadastro

## 5.4 Testador 4

- **Task 01:** Login
- **Task 05:** Recuperação de Senha

## 6. CONSIDERAÇÕES FINAIS

O Sistema de Gestão de Projetos Integradores (SGPI) foi desenvolvido com foco na gestão acadêmica da FATEC Zona Leste, especificamente para o curso de Desenvolvimento de Software Multiplataforma. O sistema contempla funcionalidades completas para diferentes perfis de usuários (aluno, professor, professor orientador, coordenador e administrador), incluindo:

- Gestão de cadastros e autenticação com sistema de palavras-chave
- Controle de disciplinas e períodos letivos
- Gerenciamento de grupos e projetos integradores
- Sistema completo de atividades com entregas e avaliações
- Sistema de reconsideração de notas
- Geração de relatórios
- Notificações por e-mail
- Recursos de acessibilidade
- Documentação legal (termos de uso e política de privacidade)

### 6.1 Tecnologias Utilizadas

#### Frontend:

- HTML5, CSS3, JavaScript (ES6+)
- Arquitetura modular com separação de responsabilidades
- UserWay Accessibility Widget

#### Backend:

- Node.js
- Arquitetura MVC
- Sistema de middlewares e helpers
- Pool de conexões otimizado
- Sistema de autenticação e autorização (RBAC)

#### Banco de Dados:

- MySQL
- Índices otimizados para performance
- Tabelas de histórico para auditoria
- Procedures e triggers

### 6.2 Padrões e Boas Práticas

O sistema foi desenvolvido seguindo padrões de mercado e boas práticas de desenvolvimento:

- Tratamento global de erros com códigos padronizados
- Validação rigorosa de entrada de dados
- Segurança com CORS, Helmet e Rate Limiting
- Logging estruturado e rastreabilidade
- Versionamento de API
- Paginação e filtragem consistentes

- Otimização de performance e cache
- Observabilidade e monitoramento de saúde do sistema
- Migrations versionadas e seeds para ambientes de desenvolvimento

O projeto foi executado de forma estruturada em 6 meses, com entregas incrementais e testes contínuos, garantindo qualidade, segurança e escalabilidade do sistema.