

Visão Geral do Sistema

Objetivo Principal

O ****Sistema de Vagas CGB Energia**** é uma solução completa para automatizar e otimizar todo o processo de recrutamento e seleção da empresa, desde a publicação de vagas até a contratação final.

Problemas Resolvidos

Antes do Sistema

- Processo manual e demorado
- Falta de padronização
- Dificuldade no acompanhamento
- Comunicação fragmentada
- Ausência de métricas
- Validações jurídicas manuais

Depois do Sistema

- Processo automatizado e ágil
- Fluxo padronizado e auditável
- Acompanhamento em tempo real
- Comunicação centralizada
- Dashboards e relatórios completos
- Validações jurídicas integradas

Principais Benefícios

Para a Empresa

- **Redução de Tempo**: Processo 60% mais rápido
- **Maior Qualidade**: Seleção mais criteriosa
- **Compliance**: Validações jurídicas obrigatórias
- **Métricas**: Dados para tomada de decisão
- **Padronização**: Processo uniforme em todas as unidades

Para o RH

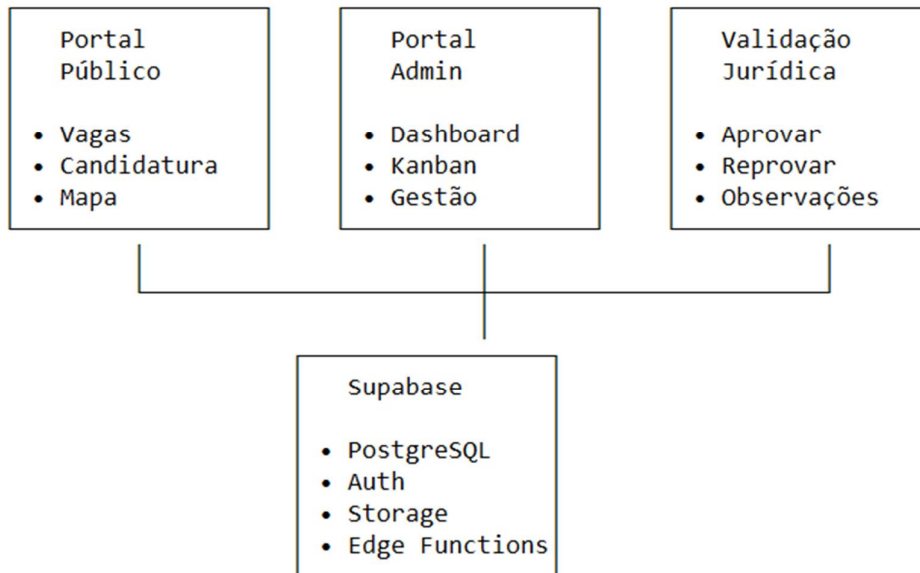
- **Centralização**: Tudo em uma plataforma
- **Automação**: Menos trabalho manual
- **Visibilidade**: Status em tempo real
- **Comunicação**: Email/WhatsApp integrados
- **Relatórios**: Métricas automáticas

Para Candidatos

- **Transparência**: Acompanhamento do processo
- **Agilidade**: Resposta mais rápida
- **Facilidade**: Candidatura online simples
- **Comunicação**: Notificações automáticas

Arquitetura Geral

...



...

Módulos Principais

1. Portal Público

****Objetivo**:** Permitir que candidatos vejam vagas e se candidatem

****Funcionalidades**:**

- Visualização de vagas em mapa interativo
- Lista de vagas com filtros
- Formulário de candidatura completo
- Upload de currículo

2. Portal Administrativo

****Objetivo**:** Gestão completa do processo seletivo

****Funcionalidades**:**

- Dashboard com métricas
- Gestão de vagas (CRUD)

- Kanban de candidatos
- Comunicação integrada
- Relatórios e analytics

3. Validação Jurídica

****Objetivo**:** Garantir compliance legal antes da contratação

****Funcionalidades**:**

- Interface específica para jurídico
- Aprovação/reprovação com comentários
- Histórico de validações
- Bloqueio automático até aprovação

4. Gestão de Usuários

****Objetivo**:** Controle de acesso e permissões

****Funcionalidades**:**

- Diferentes roles (Admin, RH, Gerência, Jurídico)
- Controle geográfico (estados/cidades)
- Validação rigorosa de senhas
- Auditoria de ações

Fluxo Básico

...

Candidato → Candidatura → Análise RH → Entrevistas → Validação TJ → Aprovação → Contratação

↓ ↓ ↓ ↓ ↓ ↓ ↓

Portal Formulário Kanban Agenda Jurídico Status HRIS

Público + Upload Admin + Email Específico Final Integração

...

Métricas e KPIs

Métricas Principais

- **Tempo Médio de Contratação**: 15 dias (meta)
- **Taxa de Conversão**: 5% candidatos → contratados
- **Satisfação do Candidato**: 4.5/5 (pesquisa)
- **Eficiência do RH**: 80% redução tempo manual

Dashboards Disponíveis

- **Funil de Conversão**: Por etapa do processo
- **Tempo por Etapa**: Identificar gargalos
- **Performance por Vaga**: Quais atraem mais candidatos
- **Análise Geográfica**: Distribuição por região

Integrações Atuais

Email

- **Provedor**: Supabase Edge Functions
- **Templates**: Personalizados por etapa
- **Automação**: Notificações automáticas

WhatsApp

- **Integração**: Web API
- **Uso**: Comunicação direta com candidatos
- **Templates**: Mensagens pré-definidas

Storage

- **Currículos**: Supabase Storage
- **Documentos**: Upload seguro
- **Backup**: Automático

Próximos Passos

Curto Prazo

- ☐ Integração com LinkedIn
- ☐ App mobile (PWA)
- ☐ Relatórios avançados
- ☐ Testes técnicos online

Médio Prazo

- ☐ IA para matching de candidatos
- ☐ Integração com HRIS
- ☐ Sistema de referências
- ☐ Analytics avançado

Longo Prazo

- ☐ Microserviços
- ☐ Event-driven architecture
- ☐ Machine Learning
- ☐ Background check automático

Funcionalidades por Módulo

Portal Público

1. Mapa Interativo de Vagas

****Localização****: Página inicial (`/ `)

****Funcionalidades****:

- Plotagem de vagas por coordenadas geográficas
- Clusters automáticos para múltiplas vagas na mesma cidade
- Popup com informações básicas da vaga
- Filtros por estado e cidade
- Zoom automático baseado na localização do usuário

****Como Usar****:

1. Acesse a página inicial
2. Visualize as vagas plotadas no mapa
3. Clique em um marcador para ver detalhes
4. Use os filtros na lateral para refinar a busca

2. Lista de Vagas

****Localização****: Página inicial (abaixo do mapa)

****Funcionalidades****:

- Grid responsivo de cards de vagas
- Informações: título, localização, tipo, data
- Filtros: cidade, estado, tipo de contrato
- Busca por texto livre
- Ordenação por data de publicação

****Campos Exibidos**:**

- Título da vaga
- Cidade e estado
- Tipo de contrato (CLT, PJ, Estágio)
- Data de publicação
- Departamento

3. Detalhes da Vaga

****Localização**:** `/job/:id`

****Funcionalidades**:**

- Descrição completa da vaga
- Requisitos obrigatórios e desejáveis
- Benefícios oferecidos
- Informações sobre a empresa
- Botão "Candidatar-se"

4. Formulário de Candidatura

****Localização**:** `/application/:jobId`

****Seções do Formulário**:**

Dados Pessoais

- Nome completo
- Email
- Telefone/WhatsApp
- Cidade e estado de residência
- Idade

Dados Profissionais

- Cargo desejado

- Já trabalhou na CGB? (Sim/Não)
- Upload de currículo (PDF, max 5MB)

Informações de Transporte

- Possui CNH? (A, B, C, D, E, Não possui)
- Possui veículo próprio?
- Modelo e ano do veículo (se aplicável)
- Disponibilidade para viagens

Informações Especiais

- Pessoa com deficiência (PCD)?
- Disponibilidade para mudança
- Pretensão salarial

LGPD

- Consentimento obrigatório para tratamento de dados
- Link para política de privacidade

Validações:

- Campos obrigatórios marcados
- Formato de email válido
- Telefone no formato brasileiro
- Arquivo PDF válido para currículo
- Consentimento LGPD obrigatório

Portal Administrativo

1. Dashboard Principal

****Localização****: `/admin`

****Acesso****: Admin, RH, Gerência

****Métricas Principais****:

- Total de vagas ativas
- Total de candidatos
- Candidatos aprovados este mês
- Tempo médio de contratação
- Taxa de conversão por etapa

****Gráficos****:

- ****Funil de Conversão****: Candidatos por etapa do processo
- ****Linha do Tempo****: Contratações nos últimos 6 meses
- ****Pizza****: Distribuição de candidatos por status
- ****Barras****: Top 5 vagas com mais candidatos

****Filtros****:

- Período (último mês, trimestre, ano)
- Estado/cidade (baseado nas permissões do usuário)
- Departamento

2. Gestão de Vagas

****Localização****: `/admin/jobs`

****Acesso****: Admin, RH

2.1 Lista de Vagas

****Funcionalidades****:

- Visualização em tabela com paginação
- Filtros: status, cidade, departamento, data
- Busca por título ou descrição
- Ações: editar, pausar, excluir, duplicar

****Colunas da Tabela****:

- Título
- Localização (cidade/estado)
- Departamento
- Tipo de contrato
- Status (ativa, pausada, finalizada)
- Data de criação
- Candidatos (quantidade)
- Ações

2.2 Criar/Editar Vaga

****Formulário Completo****:

****Informações Básicas****:

- Título da vaga
- Departamento
- Cidade e estado
- Tipo de contrato
- Carga horária

****Descrição****:

- Descrição detalhada da função
- Principais responsabilidades
- Perfil ideal do candidato

****Requisitos**:**

- Requisitos obrigatórios (array)
- Requisitos desejáveis (array)
- Experiência mínima necessária

****Benefícios**:**

- Lista de benefícios oferecidos
- Informações sobre salário (opcional)

****Configurações**:**

- Status da vaga
- Data de expiração
- Visibilidade (pública/interna)

3. Gestão de Candidatos

****Localização**:** ` /admin/candidates`

****Acesso**:** Admin, RH, Gerência

3.1 Lista de Candidatos

****Funcionalidades**:**

- Tabela com paginação e busca
- Filtros avançados por múltiplos critérios
- Exportação para CSV/Excel
- Ações em lote (mover status, enviar email)

****Filtros Disponíveis**:**

- Status do processo
- Vaga aplicada
- Cidade/estado
- Idade

- CNH
- PCD
- Data de aplicação

****Colunas da Tabela**:**

- Nome
- Email
- Telefone
- Vaga aplicada
- Status atual
- Data de aplicação
- Tempo no status atual
- Ações

3.2 Detalhes do Candidato

****Modal Completo com Abas**:**

****Aba: Detalhes****

- Informações pessoais completas
- Dados de contato
- Informações da vaga aplicada
- Dados de transporte
- Link para download do currículo

****Aba: Histórico****

- Timeline completa de todas as ações
- Mudanças de status com data/hora
- Comunicações enviadas
- Validações jurídicas com comentários
- Notas adicionadas pela equipe

****Aba: Comunicação****

- Envio de emails com templates
- Envio de mensagens WhatsApp
- Histórico de comunicações
- Templates personalizáveis

4. Processos Seletivos (Kanban)

****Localização****: `/admin/selection`

****Acesso****: Admin, RH

4.1 Interface Kanban

****Funcionalidades****:

- Drag & Drop entre colunas
- Filtro por vaga específica
- Abas: Ativos, Reprovados, Contratados
- Contador de candidatos por coluna
- Tempo em cada etapa

****Colunas do Processo****:

1. ****Cadastrado****: Candidatos recém-inscritos
2. ****Análise de Currículo****: Triagem inicial
3. ****Pré-selecionado****: Aprovados na triagem
4. ****Entrevista com RH****: Primeira entrevista
5. ****Entrevista com Gestor****: Entrevista técnica
6. ****Teste Técnico****: Avaliação específica
7. ****Aguardando Retorno****: Decisão pendente
8. ****Validação TJ****: Análise jurídica
9. ****Validação Frota****: Aprovação final
10. ****Aprovado****: Candidato selecionado
11. ****Contratado****: Processo finalizado
12. ****Reprovado****: Não selecionado

4.2 Regras de Negócio

****Bloqueios Automáticos**:**

- Candidatos em "Validação TJ" só podem sair após aprovação jurídica
- Movimentação para "Reprovado" exige motivo obrigatório
- Histórico automático de todas as movimentações

4.3 Cards dos Candidatos

****Informações Exibidas**:**

- Avatar com iniciais
- Nome do candidato
- Tempo na etapa atual
- Ícone de status jurídico (quando aplicável)

5. Gestão de Usuários RH

****Localização**:** ` /admin/users `

****Acesso**:** Admin

5.1 Lista de Usuários

****Funcionalidades**:**

- Visualização de todos os usuários RH
- Filtros por role e status
- Busca por nome ou email
- Ações: editar, desativar, resetar senha

5.2 Criar/Editar Usuário

****Formulário**:**

- Nome completo
- Email corporativo
- Role (Admin, RH, Gerência, Jurídico)
- Estados atribuídos

- Cidades atribuídas
- Status (ativo/inativo)

****Validação de Senha Rigorosa**:**

- Mínimo 8 caracteres
- Pelo menos 1 letra minúscula
- Pelo menos 1 letra maiúscula
- Pelo menos 1 número
- Pelo menos 1 caractere especial
- Confirmação obrigatória

6. Validação Legal

****Localização**:** `/admin/legal`

****Acesso**:** Jurídico

6.1 Interface Específica

****Funcionalidades**:**

- Lista de candidatos aguardando validação
- Visualização completa dos dados do candidato
- Informações detalhadas da vaga
- Seção expansível com dados completos

6.2 Dados Exibidos

****Informações Principais**:**

- Nome e dados de contato
- Vaga aplicada (título, cidade, departamento)
- Data de aplicação

****Detalhes Expandidos**:**

- ****Dados da Vaga**:** Tipo, carga horária, descrição
- ****Dados Pessoais**:** Idade, PCD, localização

- **Transporte**: CNH, veículo, modelo/ano
- **Histórico CGB**: Se já trabalhou na empresa

6.3 Ações de Validação

Opções Disponíveis:

- **Aprovar**: Libera para próxima etapa
- **Reprovar**: Bloqueia com motivo obrigatório
- **Aprovar com Restrição**: Aprova com observações

Funcionalidades:

- Comentários obrigatórios para reprovação/restrrição
- Modal de confirmação com resumo
- Histórico automático de todas as validações

7. Banco de Talentos

Localização: ` /admin/talent-bank`

Acesso: Admin, RH

Funcionalidades:

- Pool de candidatos aprovados
- Busca avançada por skills
- Filtros por localização e experiência
- Convite para novas vagas
- Histórico de participações

8. Relatórios e Analytics

Localização: ` /admin/reports`

Acesso: Admin, RH, Gerência

8.1 Relatórios Disponíveis

- ****Funil de Conversão****: Por período e vaga
- ****Tempo por Etapa****: Identificação de gargalos
- ****Performance por Vaga****: Métricas detalhadas
- ****Análise Geográfica****: Distribuição regional
- ****Eficiência do RH****: Produtividade da equipe

8.2 Exportação

- Formato CSV para análise externa
- Filtros personalizáveis
- Agendamento de relatórios automáticos

Controle de Acesso

Roles e Permissões

Admin

- Acesso total a todas as funcionalidades
- Gestão de usuários
- Configurações do sistema
- Sem restrições geográficas

RH

- Gestão de candidatos
- Processos seletivos
- Comunicação
- Relatórios básicos
- Limitado por territórios atribuídos

Gerência

- Visualização de candidatos
- Relatórios
- Aprovação de entrevistas
- Sem edição de dados

Jurídico

- Validação legal exclusiva
- Histórico de validações
- Sem restrições geográficas
- Sem acesso a outras funcionalidades

Responsividade

Todas as funcionalidades são **100% responsivas**:

- **Mobile**: Interface otimizada para smartphones
- **Tablet**: Layout adaptado para tablets
- **Desktop**: Experiência completa

Documentação atualizada em: 27/12/2024

Troubleshooting - Guia de Resolução de Problemas

Problemas Críticos

1. Sistema Fora do Ar

Sintomas:

- Página não carrega
- Erro 500/503
- Timeout de conexão

Diagnóstico:

```bash

# Verificar status do Supabase

curl -I https://[seu-projeto].supabase.co/rest/v1/

# Verificar logs do Vercel/Netlify

vercel logs --follow

# Verificar status da aplicação

curl -I https://[seu-dominio].com/health

```

Soluções:

1. ****Supabase Down****: Aguardar restauração ou usar backup
2. ****Deploy com Bug****: Rollback para versão anterior
3. ****Limite de Conexões****: Reiniciar conexões do banco

2. Banco de Dados Inacessível

Sintomas:

- Erro "connection refused"
- Queries falhando
- Timeout de database

Diagnóstico:

` `` sql

-- Verificar conexões ativas

```
SELECT count(*) FROM pg_stat_activity;
```

-- Verificar queries lentas

```
SELECT query, query_start, state
```

```
FROM pg_stat_activity
```

```
WHERE state = 'active'
```

```
AND query_start < NOW() - INTERVAL '5 minutes';
```

-- Verificar locks

```
SELECT * FROM pg_locks WHERE NOT granted;
```

` ``

Soluções:

1. **Muitas Conexões**: Implementar connection pooling
2. **Queries Lentas**: Otimizar queries ou adicionar índices
3. **Deadlocks**: Revisar ordem de operações

Problemas Comuns

1. Login Não Funciona

Problema: Usuário não consegue fazer login

```

**Diagnóstico**:
```typescript
// Verificar se usuário existe
const { data: user } = await supabase.auth.getUser();
console.log('User:', user);

// Verificar perfil RH
const { data: rhProfile } = await supabase
 .from('rh_users')
 .select('*')
 .eq('user_id', user?.id)
 .single();
console.log('RH Profile:', rhProfile);
```

```

- **Soluções**:**
- Verificar se email está confirmado
 - Resetar senha se necessário
 - Verificar se perfil RH existe
 - Conferir permissões de acesso

2. Candidato em Validação TJ Não Bloqueia

Problema: Candidato sai de "Validação TJ" sem aprovação

```

**Diagnóstico**:
```sql
-- Verificar status legal do candidato
SELECT id, name, status, legal_status
FROM candidates

```

```
WHERE status = 'Validação TJ';
```

```
-- Verificar validações existentes
```

```
SELECT * FROM candidate_legal_validations
```

```
WHERE candidate_id = '[candidate-id]'
```

```
ORDER BY created_at DESC;
```

```
```
```

```
**Soluções**:
```

- Verificar se `legal_status` está como "pendente"
- Conferir lógica de bloqueio no Kanban
- Resetar status legal se necessário:

```
```sql
```

```
UPDATE candidates
```

```
SET legal_status = 'pendente'
```

```
WHERE id = '[candidate-id]' AND status = 'Validação TJ';
```

```
```
```

3. Upload de Currículo Falha

Problema: Erro ao fazer upload de PDF

```
**Diagnóstico**:
```

```
```typescript
```

```
// Verificar configurações do Storage
```

```
const { data: buckets } = await supabase.storage.listBuckets();
```

```
console.log('Buckets:', buckets);
```

```
// Verificar políticas RLS
```

```
const { data: policies } = await supabase
```

```
.from('storage.policies')
.select('*');
` ``
```

**\*\*Soluções\*\*:**

- Verificar tamanho do arquivo (max 5MB)
- Confirmar formato PDF
- Verificar políticas do Supabase Storage
- Conferir permissões de upload

### ### 4. Emails Não Enviados

#### Problema: Sistema não envia emails

**\*\*Diagnóstico\*\*:**

```
` `` typescript
// Testar Edge Function
const { data, error } = await supabase.functions.invoke('send-email', {
 body: {
 to: 'test@example.com',
 subject: 'Teste',
 html: '<p>Teste</p>'
 }
});

console.log('Email result:', { data, error });
` ``
```

**\*\*Soluções\*\*:**

- Verificar configuração da Edge Function
- Conferir variáveis de ambiente



- Validar formato do email
- Verificar logs do Supabase

### ### 5. Performance Lenta

#### #### Problema: Sistema lento para carregar

##### **\*\*Diagnóstico\*\*:**

```
` `` `sql
```

```
-- Queries mais lentas
```

```
SELECT query, mean_exec_time, calls
```

```
FROM pg_stat_statements
```

```
ORDER BY mean_exec_time DESC
```

```
LIMIT 10;
```

```
-- Índices não utilizados
```

```
SELECT schemaname, tablename, indexname, idx_scan
```

```
FROM pg_stat_user_indexes
```

```
WHERE idx_scan = 0;
```

```
` `` `
```

##### **\*\*Soluções\*\*:**

- Adicionar índices necessários
- Otimizar queries N+1
- Implementar paginação
- Usar cache quando apropriado

```

```

## ## Ferramentas de Diagnóstico

### ### 1. Health Check Endpoint

```
````typescript
// /api/health
export async function GET() {
  const checks = {
    database: await checkDatabase(),
    storage: await checkStorage(),
    email: await checkEmailService(),
    external_apis: await checkExternalAPIs(),
  };

  const allHealthy = Object.values(checks).every(check => check.status === 'ok');

  return Response.json({
    status: allHealthy ? 'healthy' : 'unhealthy',
    timestamp: new Date().toISOString(),
    checks
  }, {
    status: allHealthy ? 200 : 503
  });
}

async function checkDatabase() {
  try {
    const { data } = await supabase.from('jobs').select('count').single();
    return { status: 'ok', latency: Date.now() };
  } catch (error) {
    return { status: 'error', error: error.message };
  }
}
```

```

### ### 2. Logs Estruturados

```typescript

```
interface LogEntry {  
  level: 'info' | 'warn' | 'error' | 'debug';  
  message: string;  
  timestamp: string;  
  userId?: string;  
  requestId?: string;  
  metadata?: Record<string, any>;  
}
```

```
class Logger {  
  static info(message: string, metadata?: any) {  
    console.log(JSON.stringify({  
      level: 'info',  
      message,  
      timestamp: new Date().toISOString(),  
      metadata  
    }));  
  }  
  
  static error(message: string, error?: Error, metadata?: any) {  
    console.error(JSON.stringify({  
      level: 'error',  
      message,  
      error: error?.stack,  
      timestamp: new Date().toISOString(),  
      metadata
```

```

    });
  }
}
```

```

### ### 3. Performance Monitoring

```

```typescript
// Hook para monitorar performance
const usePerformanceMonitor = () => {
  useEffect(() => {
    const observer = new PerformanceObserver((list) => {
      list.getEntries().forEach((entry) => {
        if (entry.entryType === 'navigation') {
          console.log('Page Load Time:', entry.loadEventEnd - entry.loadEventStart);
        }

        if (entry.entryType === 'measure') {
          console.log('Custom Measure:', entry.name, entry.duration);
        }
      });
    });

    observer.observe({ entryTypes: ['navigation', 'measure'] });

    return () => observer.disconnect();
  }, []);
};
```

```

---

## ## Monitoramento Proativo

### ### 1. Alertas Configurados

```typescript

```
interface AlertRules {  
  critical: {  
    database_down: {  
      condition: 'connection_failed > 0';  
      notification: 'immediate_call';  
    };  
  
    high_error_rate: {  
      condition: 'error_rate > 5%';  
      notification: 'slack_urgent';  
    };  
  };  
  
  warning: {  
    slow_queries: {  
      condition: 'avg_query_time > 2s';  
      notification: 'slack_warning';  
    };  
  
    high_memory_usage: {  
      condition: 'memory_usage > 80%';  
      notification: 'email';  
    };  
  };  
}
```

```

## ### 2. Métricas Importantes

```
```typescript
const metrics = {
  // Performance
  response_time: 'avg(response_time) by endpoint',
  error_rate: 'count(errors) / count(requests) * 100',
  throughput: 'count(requests) per minute',

  // Business
  daily_applications: 'count(new_applications) per day',
  conversion_rate: 'count(hired) / count(applied) * 100',
  time_to_hire: 'avg(days_from_application_to_hire)',

  // System
  database_connections: 'count(active_connections)',
  memory_usage: 'memory_used / memory_total * 100',
  disk_usage: 'disk_used / disk_total * 100',
};
```
```

---

---

## ## Otimizações de Performance

### ### 1. Database Query Optimization

```
```sql
-- Índices recomendados para performance
```

```
CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_candidates_job_status
ON candidates(job_id, status)
WHERE status != 'Reprovado';
```

```
CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_candidates_legal_pending
ON candidates(legal_status, updated_at)
WHERE legal_status = 'pendente';
```

```
CREATE INDEX CONCURRENTLY IF NOT EXISTS idx_notes_candidate_recent
ON candidate_notes(candidate_id, created_at DESC);
```

```
-- Query otimizada para dashboard
```

```
CREATE MATERIALIZED VIEW dashboard_metrics AS
```

```
SELECT
```

```
    COUNT(*) FILTER (WHERE status = 'ativa') as active_jobs,
```

```
    COUNT(*) FILTER (WHERE created_at > NOW() - INTERVAL '30 days') as
new_candidates_month,
```

```
    AVG(EXTRACT(EPOCH FROM (hired_date - applied_date))/86400) as avg_time_to_hire
```

```
FROM jobs j
```

```
LEFT JOIN candidates c ON j.id = c.job_id;
```

```
-- Refresh automático da view
```

```
CREATE OR REPLACE FUNCTION refresh_dashboard_metrics()
```

```
RETURNS void AS $$
```

```
BEGIN
```

```
    REFRESH MATERIALIZED VIEW dashboard_metrics;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
--
```

```
### 2. Frontend Optimization
```

```
` `` typescript
```

```
// Lazy loading de componentes
```

```
const CandidateDetailModal = lazy(() => import('./CandidateDetailModal'));
```

```
const ReportsPage = lazy(() => import('../pages/Reports'));
```

```
// Memoização de componentes pesados
```

```
const CandidateList = memo(({ candidates, filters }) => {
```

```
  const filteredCandidates = useMemo(() =>
```

```
    candidates.filter(candidate =>
```

```
      applyFilters(candidate, filters)
```

```
    ), [candidates, filters]
```

```
  );
```

```
  return (
```

```
    <VirtualizedList
```

```
      items={filteredCandidates}
```

```
      renderItem={CandidateCard}
```

```
      height={600}
```

```
    />
```

```
  );
```

```
});
```

```
// Debounce para busca
```

```
const useDebounce = (value: string, delay: number) => {
```

```
  const [debouncedValue, setDebouncedValue] = useState(value);
```

```
  useE  ect(() => {
```

```
    const handler = setTimeout(() => {
```

```
      setDebouncedValue(value);
```

```
    }, delay);
```



```
    return () => clearTimeout(handler);  
  }, [value, delay]);
```

```
    return debouncedValue;  
  };  
  ...  
  
  ...
```

Contatos de Emergência

Equipe Técnica

- **Desenvolvedor Principal**: +55 93 991644916

Fornecedores

- **Supabase Support**: support@supabase.io

- **Vercel Support**: support@vercel.com

Procedimentos de Emergência

Severidade 1 (Sistema Fora do Ar)

1. Ligar imediatamente para desenvolvedor principal
2. Enviar email para equipe técnica
3. Postar no Slack #emergencia
4. Documentar incidente

Severidade 2 (Funcionalidade Crítica Afetada)

1. Postar no Slack #suporte
2. Enviar email com detalhes

3. Criar ticket de suporte

Severidade 3 (Problema Menor)

1. Enviar email para suporte
2. Documentar no sistema de tickets

Recursos Úteis

Documentação Externa

- [Supabase Docs](https://supabase.com/docs)
- [React Query Docs](https://tanstack.com/query/latest)
- [Tailwind CSS Docs](https://tailwindcss.com/docs)

Ferramentas de Debug

- **React DevTools**: Para debug de componentes
- **TanStack Query DevTools**: Para debug de queries
- **Supabase Dashboard**: Para debug de banco de dados

Logs e Monitoramento

- **Supabase Logs**: Dashboard do Supabase
- **Vercel Analytics**: Métricas de performance
- **Browser DevTools**: Network, Console, Performance

Guia de Troubleshooting atualizado em: 27/12/2024