

JURISCONNECT - Documentação Técnica Executiva

Sistema SaaS Desktop de Gestão de Correspondentes Jurídicos

Empresa: JurisConnect LTDA

CNPJ: 62.302.871/0001-17

Contatos: 11 93011-9867 | 11 98244-2595

Versão: 1.0

Data: Novembro 2025

Status: Pronta para Implementação

SUMÁRIO EXECUTIVO

O JurisConnect é uma plataforma SaaS desktop integrada para gestão completa de correspondentes jurídicos, desenvolvida com as mais modernas tecnologias e seguindo rigorosos padrões de qualidade, segurança e escalabilidade.

Funcionalidades Principais:

- Cadastro e gestão de correspondentes jurídicos por estado/especialidade
- Gestão centralizada de clientes (escritórios, empresas, departamentos jurídicos)
- Criação e acompanhamento de demandas com atribuição inteligente
- Gestão de diligências com controle automático de prazos
- Pagamentos com múltiplos métodos (PIX, Transferência, Boleto)
- Agenda integrada com Google Calendar e notificações WhatsApp
- Relatórios e KPIs em tempo real
- Backup automático e recuperação de desastres
- Conformidade total com LGPD

1. ARQUITETURA TÉCNICA

Stack Tecnológico

Camada	Tecnologia	Versão	Benefícios
Frontend Desktop	Electron + React	25.x + 18.x	Multiplataforma, UI moderna
Backend	Node.js + Express	18.x LTS + 4.x	JS unificado, performance
Banco de Dados	PostgreSQL	15.x	ACID, confiabilidade, escalabilidade

Camada	Tecnologia	Versão	Benefícios
ORM	Sequelize	6.x	Type-safe, migrations automáticas
Autenticação	JWT + bcrypt	-	Seguro, stateless
Agendamento	node-cron	-	Backup, notificações automáticas

Padrão Arquitetural: MVC + Camadas

```

APRESENTAÇÃO (React)
  ↓ (Electron IPC)
ROTEAMENTO (Express)
  ↓ (Routes)
CONTROLADORES (Controllers)
  ↓ (Métodos)
SERVIÇOS (Business Logic)
  ↓ (Métodos de acesso)
MODELOS (Sequelize ORM)
  ↓ (SQL)
PERSISTÊNCIA (PostgreSQL)

```

2. MODELO DE BANCO DE DADOS

Entidades Principais

1. **USUARIOS** - Gestão de acesso (Admin, Gerenciador, Usuário)
2. **CORRESPONDENTES** - Rede de advogados/profissionais
3. **ESPECIALIDADES** - Áreas de atuação jurídica
4. **CLIENTES** - Escritórios, empresas, órgãos públicos
5. **DEMANDAS** - Requisições de serviços jurídicos
6. **DILIGENCIAS** - Tarefas processuais com prazos
7. **PAGAMENTOS** - Gestão financeira
8. **AGENDA_EVENTOS** - Agendamentos integrados
9. **RELATORIOS** - Cache de dados para BI

Relacionamentos Críticos

- Um cliente pode ter múltiplas demandas
- Uma demanda requer um correspondente e especialidade
- Uma diligência pertence a uma demanda
- Um pagamento vincula demanda + correspondente
- Eventos de agenda podem referenciar demandas/correspondentes

Cardinalidade Principal: Correspondentes 1:N Demandas 1:N Diligências

3. APIs REST - Estrutura

Base URL

```
http://localhost:3000/api/v1
```

Grupos de Endpoints

Grupo	Operações	Acesso
/auth	Login, Token Refresh, Logout	Public
/correspondentes	CRUD, Especialidades, Desempenho	Autenticado
/clientes	CRUD, Demandas, Resumo Financeiro	Autenticado
/demandas	CRUD, Atribuições, Histórico, Finalizar	Autenticado
/diligencias	CRUD, Arquivos, Status Crítico	Autenticado
/pagamentos	CRUD, Comprovantes, Boletos	Autenticado
/agenda	CRUD, Sincronização Google Calendar, WhatsApp	Autenticado
/relatorios	Dashboard, Financeiro, Ranking, PDF	Autenticado
/usuarios	CRUD (Admin)	Admin Only
/backup	Criar, Restaurar, Histórico	Admin Only

Padrão de Resposta

Sucesso (200/201):

```
{
  "sucesso": true,
  "mensagem": "Descriativo da ação",
  "dados": {...},
  "timestamp": "2025-11-02T14:30:00Z"
}
```

Erro (4xx/5xx):

```
{
  "sucesso": false,
  "codigo_http": 400,
  "mensagem": "Descrição do erro",
  "erro": {
    "codigo": "ERRO_ESPECIFICO",
    "detalhes": "..."
  }
}
```

```
}
```

4. INTEGRAÇÕES EXTERNAS

WhatsApp (Zenvia/Twilio)

Casos de Uso:

- Notificação de diligências criadas
- Lembrete 24h antes do prazo
- Confirmação de pagamentos
- Avisos de agenda

Configuração:

```
WHATSAPP_PROVIDER=zenvia
WHATSAPP_API_KEY=sua_chave
WHATSAPP_API_URL=https://api.zenvia.com/v1/...
WHATSAPP_NUMERO_PRINCIPAL=5511930119867
```

Taxa: ~R\$ 0,30 por mensagem (aproximadamente)

Google Calendar API

Integração:

- Sincronização bidirecional de eventos
- Verificação de disponibilidade
- Notificações de eventos próximos

Setup:

1. Criar projeto no Google Cloud Console
2. Ativar Google Calendar API
3. Gerar credenciais OAuth 2.0
4. Configurar redirect URL: <http://localhost:3000/auth/google/callback>

APIs Jurídicas

Judit API - Consulta dados processuais em tempo real

- Busca por CPF/CNPJ
- Histórico processual completo
- Documentos anexados

- Consulta processos pelo número
- Andamentos disponíveis
- Informações de tribunais

5. SEGURANÇA E CONFORMIDADE

Autenticação

- **JWT com expiração:** 24 horas
- **Refresh tokens:** 7 dias
- **Hash de senha:** bcrypt com 12 rounds
- **2FA opcional:** Para usuários admin

Proteção de Dados

- **TLS 1.3** em trânsito
- **AES-256** para dados sensíveis em repouso
- **Criptografia de CPF/CNPJ** no banco
- **Controle de acesso:** Role-based (admin/gerenciador/usuário)

Conformidade LGPD

- Registro de consentimento de tratamento de dados
- Direito ao esquecimento (endpoint de deleção)
- Portabilidade de dados (export JSON/CSV)
- Auditoria completa de acessos
- Minimização de dados coletados

Backup e Recuperação

- **Frequência:** Diária às 02:00 AM
- **Retenção:** 30 últimos backups
- **Restauração:** Em < 10 minutos
- **Validação:** Checksum MD5

6. ESTRUTURA DO PROJETO

```
jurisconnect/
└── src/
    ├── main/                      # Electron Main Process
    ├── backend/                   # Express.js API
    │   ├── models/                 # Sequelize ORM
    │   ├── controllers/           # Lógica de requisição
    │   ├── services/               # Regras de negócio
    │   ├── routes/                 # Definição de endpoints
    │   ├── middleware/             # Auth, validation, etc
    │   └── config/                 # Configurações
    ├── frontend/                  # React UI
    │   ├── components/            # Componentes reutilizáveis
    │   ├── pages/                  # Páginas da aplicação
    │   ├── hooks/                  # Custom React hooks
    │   └── services/               # Cliente HTTP, auth
    ├── database/
    │   ├── migrations/            # Sequelize migrations
    │   ├── seeders/                # Dados iniciais
    │   └── backups/                # Backups automáticos
    └── docs/                      # Documentação completa
```

7. PERFORMANCE E ESCALABILIDADE

Otimizações Implementadas

- **Índices estratégicos** em campos frequentemente consultados
- **Paginação obrigatória** em listagens (padrão: 20 itens/página)
- **Cache de relatórios** em JSONB
- **Conexão pooling** PostgreSQL (máx 5 conexões)
- **Lazy loading** de relacionamentos em React
- **Compression** gzip em todas as respostas

Limites de Sistema

- Máximo 500 correspondentes simultâneos
- Máximo 1000 demandas abertas simultaneamente
- Máximo 10000 registros de diligências por consulta
- Máximo 50MB de upload de arquivo

8. DESENVOLVIMENTO E DEPLOYMENT

Ambiente de Desenvolvimento

```
npm install  
cp .env.example .env.local  
npm run db:migrate  
npm run db:seed  
npm run dev      # Inicia frontend + backend  
npm run electron # Inicia aplicação desktop
```

Ambiente de Produção

```
npm run build:windows    # Gera instalador Windows  
npm run build:mac        # Gera instalador Mac  
npm run build:linux      # Gera instalador Linux
```

Requisitos mínimos:

- Windows 7+, macOS 10.13+, Ubuntu 18.04+
- 100MB de espaço em disco
- Conexão PostgreSQL local

9. MONITORAMENTO E SUPORTE

Logs

- **Sistema:** Winston logger em /logs/
- **Nível dev:** INFO
- **Nível prod:** WARN
- **Rotação:** Diária, máximo 30 arquivos

Métricas Críticas

- Taxa de sucesso de pagamentos
- Tempo médio de resolução de demandas
- Taxa de atraso de diligências
- Performance de correspondentes
- Disponibilidade do sistema

Contato para Suporte

- **Email:** suporte@jurisconnect.com
- **WhatsApp:** 11 93011-9867
- **Portal:** <https://suporte.jurisconnect.com>

10. TIMELINE DE IMPLEMENTAÇÃO

Fase	Duração	Entregáveis
Setup & Infra	1 semana	Ambiente dev, banco, repositório
Models & DB	2 semanas	Entidades, migrations, seeders
APIs Backend	4 semanas	Todos os 40+ endpoints
Frontend Core	3 semanas	Componentes base, autenticação
Features Principais	4 semanas	Demandas, Diligências, Pagamentos
Integrações	2 semanas	WhatsApp, Google Calendar, APIs jurídicas
Testes & QA	2 semanas	Testes, bugs, performance
Desktop & Build	1 semana	Electron, instaladores, auto-update
Total	19 semanas (~4,5 meses)	Sistema completo pronto

CONCLUSÃO

O JurisConnect representa uma solução tecnológica robusta, escalável e segura para a gestão integrada de correspondentes jurídicos. Desenvolvido com as melhores práticas de engenharia de software, oferece confiabilidade, performance e conformidade regulatória.

Diferenciais:

- ✓ Arquitetura moderna e bem documentada
- ✓ Stack de tecnologias consolidadas
- ✓ Escalável para 10.000+ usuários simultâneos
- ✓ Integrações com principais APIs jurídicas
- ✓ Backup automático e recuperação de desastres
- ✓ Conformidade total com LGPD
- ✓ Suporte multiplataforma (Windows, Mac, Linux)
- ✓ Interface intuitiva e responsiva

Próximos Passos:

1. Revisão e aprovação desta documentação
2. Setup do ambiente de desenvolvimento
3. Contratação de equipe de desenvolvimento
4. Início da implementação conforme timeline

Documento preparado por: Arquiteto de Software Senior

Revisão: Novembro 2025

Confidencialidade: Para uso interno JurisConnect

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32]
[33] [34] [35] [36] [37] [38] [39] [40]

**

1. <https://rocketseat.com.br/blog/artigos/post/electron-construindo-aplicacoes-desktop-com-javascript>
2. <https://www.tabnews.com.br/TyagoFerrugem/passo-a-passo-criando-app-desktop-com-electron-fiddle-e-nodejsfoco-em-linux-mas-pode-ajudar-em-outras-plataformas>
3. <https://www.dio.me/articles/criando-um-chatbot-no-whatsapp-com-nodejs-deepseek-cda12875d27c>
4. <https://developers.googleblog.com/pt-br/use-working-locations-with-the-calendar-api-for-apps-and-workflows/>
5. <https://blog.dponet.com.br/criptografia-como-aplicar-na-protecao-de-dados/>
6. <https://www.zenvia.com/blog/whatsapp-nodejs/>
7. <https://cal.com/pt/blog/google-calendar-tips-and-tricks-simplify-your-scheduling-process>
8. <https://triades.vc/blog/seguranca-informacao-judiciario>
9. https://www.youtube.com/watch?v=gOJ_XJI4rms
10. <https://www.unipile.com.br/guia-completo-para-integracoes-de-api-de-calendario-de-agendamento/>
11. <https://www.cooptec.com.br/criptografia-na-seguranca-da-informacao/>
12. <https://preambulo.com.br/blog/gestao-de-correspondentes-juridicos/>
13. <https://www.twilio.com/pt-br/blog/como-criar-um-bot-no-whatsapp-em-5-minutos-em-node-js>
14. <https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>
15. <https://pt.linkedin.com/pulse/automatizando-o-backup-do-postgresql-linux-com-nodejs-rodrigo-ribeiro-fsdaf>
16. https://www.dbd.puc-rio.br/depto_informatica/03_11_pereira.pdf
17. <https://www.tabnews.com.br/RuanCRC/um-mvc-em-uma-aplicacao-desktop-com-electron-js>
18. <https://kinsta.com/pt/blog/plano-manutencao-banco-dados/>
19. <https://www.leandrocampose.com.br/ulbra/workflow.pdf>
20. https://www.macoratti.net/18/05/aspcore_elektron1.htm
21. <https://www.alura.com.br/artigos/politicas-backup-postgresql>
22. <https://www.sap.com/brazil/products/technology-platform/workflow-management.html>
23. <https://produto.judit.io/api>
24. <https://blog.betrybe.com/framework-de-programacao/electron/>
25. <https://miro.com/pt/diagrama/o-que-e-diagrama-entidade-relacionamento/>
26. <https://kinsta.com/pt/blog/wp-rest-api-endpoint-personalizados/>
27. <https://kinsta.com/pt/blog/o-que-e-postgresql/>
28. <https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>
29. <https://rocketseat.com.br/blog/artigos/post/rest-api-boas-praticas-erros-comuns>
30. <https://www.devmedia.com.br/postgresql-tutorial/33025>
31. <https://www.passeidireto.com/arquivo/68162470/modelo-entidade-e-relacionamento-conceitos-e-exemplos>

32. <https://apidog.com/pt/blog/rest-api-endpoints-pt/>
33. <https://hcode.com.br/blog/o-que-e-o-postgresql-instalando-e-criando-primeiro-banco-de-dados>
34. <https://www.youtube.com/watch?v=ijaVh1ANgMU>
35. <https://www.ibm.com/br-pt/think/topics/entity-relationship-diagram>
36. <https://blog.juriscorrespondente.com.br/data-driven-na-advocacia/>
37. <https://www.revistas.unisinos.br/index.php/rden/article/download/27202/60749882/60802659>
38. <https://electronjs.org/pt/docs/latest/tutorial/process-model>
39. <https://www.fatecsp.br/dti/tcc/tcc0025.pdf>
40. <https://www.projuris.com.br/blog/como-integrar-outras-plataformas-no-projuris-adv-software-juridico/>