Guia de Integração da API Cailun no Sistema

1. Objetivo

Integrar o sistema atual (React + Node.js/Express) ao **GED Cailun**, permitindo:
- Criar signatários automaticamente para usuários com **assinatura digital**;
- Fazer upload de documentos;
- Associar documentos aos signatários para assinatura;
- Gerenciar **token de autenticação** com o Cailun de forma segura (apenas no backend).

2. Fluxo Arquitetural

(Representação em diagrama de sequência, descrito no documento original)

3. Estrutura do Projeto

src/
services/
cailunService.js # Gestão de token e chamadas genéricas
routes/
usuarios.js # Cadastro de usuário + signatário
documentos.js # Upload de documento + assinatura
db/ # Integração ORM (Prisma/Sequelize/Knex)
app.js # Configuração do Express

4. Gestão de Token (Backend)

- **Nunca expor o token ao frontend**.
- Token é armazenado no **servidor** (memória ou Redis).
- Sempre que expira → servidor faz login automático no Cailun.
- Implementar um *service* responsável por:
- loginCailun(): autenticar e salvar token.
- getToken(): recuperar token válido.
- callCailun(path, method, body): chamar endpoints genéricos.

5. Cadastro de Usuário

Regras

- Campo usaAssinaturaDigital no cadastro.
- Se true, cria signatário no Cailun (/signatories).
- Salvar signatoryId retornado na base de dados.

Exemplo Prompt para Controller

POST /api/usuarios

{

"nome": "João Silva",

"email": "joao@email.com",

"senha": "123456",

"usaAssinaturaDigital": true,

"cpf": "12345678900",

"phone": "11999999999"

}

Resposta:

{

"id": 1,

"nome": "João Silva",

"usaAssinaturaDigital": true,

"signatoryId": 55

}

6. Upload de Documento

Regras

- Apenas usuários com usaAssinaturaDigital = true podem assinar.
- Se usuário não tiver signatoryId → criar no momento do upload.
- Passos:

1. Upload do documento (/documents).
2. Associar signatário (/documents/{id}/signatories).

Exemplo Prompt para Controller

POST /api/documentos

{
"usuarioId": 1,
"titulo": "Contrato de Serviço",
"conteudoBase64": "JVBERi0xLjQKJ...."
}

Resposta:
{
"documentoId": 1234,
"status": "aguardando_assinatura"
}

7. Segurança

- **Cookies HTTP-only** continuam para login de usuários.
- **Credenciais do Cailun** ficam em .env
CAILUN_EMAIL=service@empresa.com.br
CAILUN_PASSWORD=senhaSuperSegura
- Sempre usar HTTPS em produção.
- Nunca expor token da Cailun ao frontend.

8. Boas práticas

- **Salvar signatoryId** no usuário local.
- Implementar **retry** em chamadas externas.
- Logar todas as operações.
- Criar testes automatizados.
- Usar Redis para cache de token em produção.

9. Roadmap de Implementação

1. Criar cailunService.js com login automático e cache de token.
2. Criar rota /api/usuarios → cadastro de usuário + criação opcional de signatário.
3. Criar rota /api/documentos → upload e vínculo com signatário.
4. Ajustar banco de dados.
5. Integrar frontend React → chamadas apenas à sua API.

6. Testar fluxo fim a fim.
7. Configurar ambiente de produção com variáveis seguras.
8. Monitorar logs e métricas.

10. Próximos passos sugeridos

- Criar um mock de integração no Postman.
- Adicionar fila assíncrona se necessário.
- Implementar dashboard interno para monitoramento.