

### ### Observations

O projeto possui uma arquitetura sólida e bem implementada, com frontend em React/TypeScript (Vercel), backend em Node.js/Express (DigitalOcean), e integração completa com SEFAZ. O sistema funciona perfeitamente em **modo simulação** e está parcialmente configurado para **homologação**, mas **não** está pronto para emissão de NFe em produção devido a configurações críticas pendentes: certificado digital não configurado, CNPJ do emitente ausente, e MongoDB de produção não configurado. A infraestrutura (Cloudflare, SSL, DNS) está 100% operacional.

### ### Approach

A estratégia para preparar o sistema para produção envolve três frentes principais: (1) **Configuração de Credenciais e Certificados** - obter e configurar certificado digital A1 válido, CNPJ da empresa e dados fiscais; (2) **Infraestrutura de Dados** - provisionar MongoDB de produção com backup automático; (3) **Validação e Testes** - testar em homologação com certificado real antes de ativar produção. O plano prioriza segurança, conformidade fiscal e rastreabilidade, seguindo as melhores práticas para sistemas de emissão de documentos fiscais.

### ### Reasoning

Analisei a estrutura completa do projeto listando diretórios recursivamente, li os arquivos de configuração de ambiente (`.env.production``, `.env.producao``, `.env.homologacao``), examinei os serviços críticos (`nfe-service.js``, `certificate-service.js``, `sefaz-client.js``), revisei a documentação do projeto (`STATUS_ATUAL.md``, `CONFIGURACAO_FINALIZADA.md``, `RELATORIO_FINAL.md``, `RESUMO_EXECUTIVO_PROJETO.md``), e identifiquei os modelos de dados (`Configuracao.js``). Busquei por variáveis críticas usando grep para confirmar o estado das configurações.

### ## Mermaid Diagram

sequenceDiagram

participant Admin as Administrador

participant Cert as Certificadora Digital

participant Server as Servidor DigitalOcean

participant DB as MongoDB Produção

participant SEFAZ as SEFAZ

Note over Admin,SEFAZ: FASE 1: PREPARAÇÃO (Dias 1-3)

Admin->>Cert: 1. Solicitar Certificado A1

Cert-->>Admin: Certificado Digital (.pfx)

Admin->>Admin: 2. Validar certificado localmente

Admin->>Server: 3. Upload certificado via SCP

Server->>Server: 4. Configurar permissões (chmod 600)

Admin->>DB: 5. Provisionar MongoDB Atlas

DB-->>Admin: String de conexão

Admin->>Server: 6. Configurar .env.production

Server->>DB: 7. Testar conectividade

DB-->>Server: Conexão OK

Admin->>Server: 8. Popular configurações no banco

Server->>DB: Inserir documento Configuracao

Admin->>Server: 9. Executar validar-producao.js

Server-->>Admin:  Todas validações OK

Note over Admin,SEFAZ: FASE 2: TESTES HOMOLOGAÇÃO (Dias 4-5)

Admin->>Server: 10. Configurar AMBIENTE=2

Admin->>Server: 11. Emitir NFe teste


Server->>Server: Gerar XML, assinar

Server->>SEFAZ: Enviar NFe (homologação)

SEFAZ-->>Server: Status 100 - Autorizado

Server->>DB: Salvar NFe

Admin->>SEFAZ: 12. Validar no portal

SEFAZ-->>Admin: NFe autorizada 

Admin->>Server: 13. Testar cancelamento

Server->>SEFAZ: Enviar evento cancelamento

SEFAZ-->>Server: Cancelamento homologado

Note over Admin,SEFAZ: FASE 3: ATIVAÇÃO PRODUÇÃO (Dia 6-7)

Admin->>Server: 14. Backup completo

Admin->>Server: 15. Alterar AMBIENTE=1

Admin->>Server: 16. Reiniciar aplicação

Server->>Server: Carregar certificado

Server->>DB: Conectar banco produção

Server->>SEFAZ: Verificar status serviço

SEFAZ-->>Server: Serviço disponível

Admin->>Server: 17. EMITIR PRIMEIRA NFE REAL

Server->>Server: Validar dados, gerar XML

Server->>Server: Assinar com certificado

Server->>SEFAZ: Enviar NFe PRODUÇÃO

SEFAZ->>SEFAZ: Validar, processar

SEFAZ-->>Server:  Status 100 - Autorizado


Server->>DB: Persistir NFe autorizada

Server->>Server: Salvar XML em /enviadas

Server-->>Admin: NFe autorizada! Chave: 44 dígitos

Admin->>SEFAZ: 18. Validar no portal SEFAZ

SEFAZ-->>Admin:  NFe válida e autorizada

Note over Admin,SEFAZ: SISTEMA EM PRODUÇÃO 

## Proposed File Changes

### backend\.env.production(MODIFY)

References:

- backend\.env.producao
- backend\.env.homologacao
- backend\.env.example

**\*\*Configurar todas as variáveis de ambiente obrigatórias para produção:\*\***

1. **\*\*MONGODB\_URI\*\***: Adicionar string de conexão do MongoDB de produção (MongoDB Atlas recomendado). Formato:  
`mongodb+srv://usuario:senha@cluster.mongodb.net/brandaocontador\_nfe`.  
Garantir que o IP do servidor DigitalOcean (159.89.228.223) esteja na whitelist do MongoDB Atlas.

2. **CNPJ\_EMITENTE**: Configurar com o CNPJ real da empresa Brandão Contabilidade (apenas números, 14 dígitos). Este CNPJ deve estar ativo na Receita Federal e autorizado para emissão de NFe no SEFAZ.
3. **CERT\_PATH**: Configurar caminho absoluto do certificado digital A1 no servidor. Recomendado: `/var/www/brandaocontador-nfe-backend/certs/certificado_producao.pfx`. O diretório `certs` deve ter permissões 700 (apenas root).
4. **CERT\_PASS**: Substituir placeholder pela senha real do certificado digital. Remover o valor `sua_senha_do_certificado_aqui` e adicionar a senha fornecida pela certificadora.
5. **JWT\_SECRET**: Gerar chave forte de 32+ caracteres aleatórios para produção. Não usar o valor atual `prod-secret-key-brandao-contador-nfe-2024`. Usar ferramenta como `openssl rand -base64 32`.
6. **UF**: Verificar se deve ser `MS` (Mato Grosso do Sul) ou manter `SVRS`. Confirmar com a empresa qual é o estado de emissão correto.
7. **SEED\_ADMIN\_SENHA**: Alterar senha padrão `@Pa2684653#` para uma senha forte única. Após primeiro acesso, comentar ou remover as variáveis `SEED_ADMIN_*` para evitar reuso.
8. **LOG\_FILE**: Verificar se o diretório `/var/www/brandaocontador-nfe-backend/logs` existe no servidor. Criar se necessário com permissões adequadas.
9. **XML\_OUTPUT\_DIR**: Adicionar variável (não presente no arquivo atual) com valor `/var/www/brandaocontador-nfe-backend/xmls` para armazenar XMLs das NFes emitidas.

10. **\*\*SAVE\_XML\_FILES\*\***: Adicionar variável com valor `true` para garantir que todos os XMLs sejam salvos (obrigatório para auditoria fiscal).

**\*\*Validações necessárias\*\***: Após configurar, testar conectividade com MongoDB, validar certificado usando o script `backend/validate-certificate-sefaz.js`, e verificar se todas as variáveis estão carregadas corretamente no ambiente.

### backend\certs(MODIFY)

References:

- backend\services\certificate-service.js
- backend\validate-certificate-sefaz.js

**\*\*Adicionar certificado digital A1 de produção ao diretório\*\***

1. **\*\*Obter Certificado\*\***: Solicitar certificado digital A1 (arquivo .pfx ou .p12) junto à certificadora autorizada (Serasa, Certisign, Valid, etc.). O certificado deve estar no nome da empresa (CNPJ) que irá emitir as NFes.

2. **\*\*Upload Seguro\*\***: Transferir o arquivo do certificado para o servidor usando SCP com conexão segura: `scp certificado\_producao.pfx root@159.89.228.223:/var/www/brandaocontador-nfe-backend/certs/`.

3. **\*\*Permissões\*\***: Configurar permissões restritivas no diretório e arquivo:

- Diretório: `chmod 700 /var/www/brandaocontador-nfe-backend/certs`
- Arquivo: `chmod 600 /var/www/brandaocontador-nfe-backend/certs/certificado\_producao.pfx`
- Owner: `chown root:root` em ambos

4. **\*\*Validação\*\***: Executar o script `backend/validate-certificate-sefaz.js` para validar o certificado antes de usar em produção. Verificar:

- Certificado não está expirado
- Certificado está no formato correto (PKCS#12)
- Senha está correta
- Certificado contém chave privada e certificado público

5. **\*\*Backup\*\***: Manter cópia segura do certificado em local externo ao servidor (cofre de senhas corporativo ou storage criptografado).

6. **\*\*Monitoramento\*\***: Configurar alerta para vencimento do certificado (recomendado: 60 dias antes). O certificado A1 tem validade de 1 ano.

**\*\*Importante\*\***: O certificado atual `MAP LTDA45669746000120.pfx` é de teste e apresenta erro 403 no SEFAZ (não autorizado). Não deve ser usado em produção. Remover ou renomear para `\_teste.pfx` para evitar confusão.

### backend\services\nfe-service.js(MODIFY)

References:

- backend\services\sefaz-client.js
- backend\assinador.js
- backend\models\Configuracao.js(MODIFY)

**\*\*Verificar e ajustar configurações para produção no serviço de NFe:\*\***

1. **\*\*Método `carregarCertificadoSistema` (linhas 40-87)\*\***: Este método já implementa a lógica correta de carregamento do certificado, com fallback para banco de dados (model `Configuracao`) e variáveis de ambiente. Verificar que:

- A variável `SIMULATION\_MODE` está definida como `false` em produção
- O certificado está sendo carregado corretamente do caminho configurado
- Logs de sucesso/erro estão sendo gerados adequadamente

2. **Método `emitirNfe` (linhas 91-173):** Validar que:

- O bloco de simulação (linhas 97-110) não será executado quando `SIMULATION\_MODE=false`
- A validação de certificado (linhas 112-120) está funcionando e retornando erro apropriado se certificado não estiver carregado
- XMLs estão sendo salvos no diretório correto antes do envio (linhas 136-140)
- Sistema de movimentação de arquivos (enviadas/falhas) está operacional (linhas 148-163)

3. **Método `gerarXmlNfe` (linhas 175-252):** Verificar que:

- O CNPJ do emitente (linha 207) está usando `this.CNPJ\_EMITENTE` corretamente
- Código da UF está sendo obtido corretamente pelo método `obterCodigoUF`
- Chave de acesso está sendo gerada com 44 dígitos (método `gerarChaveAcesso`)

4. **Método `enviarParaSefaz` (linhas 679-703):** Confirmar que:

- `SefazClient` está sendo inicializado com as configurações corretas de produção
- Timeout está adequado (30000ms padrão, configurável via env)
- Tratamento de resposta do SEFAZ está normalizado corretamente

5. **Validação de Segurança:** O método `validarXmlSeguranca` (linhas 794-816) está comentado na linha 133. Considerar reativar em produção para validar que o CNPJ no XML corresponde ao CNPJ configurado, prevenindo emissão acidental com dados incorretos.



**\*\*Não é necessário modificar código\*\*, apenas validar que as configurações de ambiente estão corretas e que o serviço está operando conforme esperado. Todos os métodos já estão implementados corretamente.**

### CHECKLIST\_PRODUCAO.md(NEW)

References:

- backend\README.md
- GUIA\_ACESSO\_ADMIN.md
- .trae\documents\RESUMO\_EXECUTIVO\_PROJETO.md

**\*\*Criar documento de checklist completo para ativação de produção:\*\***

Este arquivo deve servir como guia passo a passo para a equipe responsável pela ativação do ambiente de produção. Incluir:

1. **\*\*Seção: Pré-requisitos\*\***

- Lista de documentos necessários (certificado digital, dados da empresa, credenciais)
- Contatos importantes (SEFAZ, certificadora, suporte técnico)
- Prazos estimados para cada etapa

2. **\*\*Seção: Configuração de Certificado\*\***

- Passo a passo para obtenção do certificado A1
- Como validar o certificado localmente
- Como fazer upload seguro para o servidor
- Como testar o certificado em homologação
- Checklist de validação (validade, formato, senha, autorização SEFAZ)

### 3. **\*\*Seção: Configuração de Banco de Dados\*\***

- Opções de provisionamento (MongoDB Atlas vs local)
- Configuração de whitelist de IPs
- String de conexão e variáveis de ambiente
- Teste de conectividade
- Configuração de backup automático
- Política de retenção de dados (mínimo 5 anos para NFe)

### 4. **\*\*Seção: Configuração de Variáveis de Ambiente\*\***

- Tabela com todas as variáveis obrigatórias e seus valores esperados
- Diferenças entre homologação e produção
- Como gerar valores seguros (JWT\_SECRET, senhas)
- Como aplicar as configurações no servidor

### 5. **\*\*Seção: Testes em Homologação\*\***

- Lista de cenários de teste obrigatórios
- Emissão de NFe simples e complexa
- Consulta, cancelamento e inutilização
- Validação de XMLs gerados
- Critérios de aceitação para cada teste

### 6. **\*\*Seção: Ativação de Produção\*\***

- Procedimento de backup pré-migração
- Comandos para alternar ambiente
- Verificações pós-ativação
- Plano de rollback em caso de problemas

#### 7. **\*\*Seção: Primeira Emissão Real\*\***

- Recomendações para primeira NFe (valor baixo, dados simples)
- Como validar no portal do SEFAZ
- O que fazer em caso de erro
- Procedimento de cancelamento se necessário

#### 8. **\*\*Seção: Monitoramento Pós-Produção\*\***

- Métricas a acompanhar nas primeiras 24h
- Logs críticos a monitorar
- Alertas a configurar
- Contatos de suporte

#### 9. **\*\*Seção: Troubleshooting\*\***

- Problemas comuns e soluções
- Erros do SEFAZ e seus significados
- Como interpretar logs
- Quando escalar para suporte

#### 10. **\*\*Seção: Checklist Final\*\***

- Lista de verificação com checkboxes para cada item crítico
- Assinaturas/aprovações necessárias
- Data de ativação planejada vs real

Formatar em Markdown com emojis para facilitar leitura e uso como documento de trabalho.

### backend\scripts\validar-producao.js(NEW)

## References:

- backend\services\certificate-service.js
- backend\config\database.js
- backend\services\sefaz-client.js
- backend\models\Configuracao.js(MODIFY)

**\*\*Criar script de validação completa do ambiente de produção:\*\***

Este script deve ser executado antes de ativar a produção para garantir que todas as configurações estão corretas. Implementar verificações para:

### 1. **\*\*Validação de Variáveis de Ambiente\*\***:

- Verificar se todas as variáveis obrigatórias estão definidas (MONGODB\_URI, CERT\_PATH, CERT\_PASS, CNPJ\_EMITENTE, JWT\_SECRET)
- Validar formato do CNPJ (14 dígitos numéricos)
- Verificar se AMBIENTE=1 (produção)
- Validar se SIMULATION\_MODE=false
- Verificar se JWT\_SECRET não é o valor padrão de exemplo

### 2. **\*\*Validação de Certificado Digital\*\***:

- Verificar se arquivo do certificado existe no caminho configurado
- Validar se o certificado pode ser carregado com a senha fornecida
- Verificar validade do certificado (não expirado, válido por pelo menos 30 dias)
- Extrair e exibir informações do certificado (titular, emissor, validade)
- Validar se o CNPJ do certificado corresponde ao CNPJ\_EMITENTE configurado

### 3. **\*\*Validação de Conectividade\*\***:

- Testar conexão com MongoDB usando a URI configurada

- Verificar se consegue criar/ler documentos no banco
- Testar conectividade com SEFAZ (status do serviço)
- Validar URLs dos webservices para a UF configurada

4. **\*\*Validação de Estrutura de Diretórios\*\***:

- Verificar se diretórios necessários existem (xmls, logs, certs)
- Validar permissões dos diretórios (escrita habilitada)
- Verificar espaço em disco disponível (mínimo 10GB recomendado)

5. **\*\*Validação de Configurações no Banco\*\***:

- Verificar se existe documento de configuração no model `Configuracao`
- Validar dados da empresa (razão social, CNPJ, endereço completo)
- Verificar configurações de NFe (série, numeração inicial)

6. **\*\*Validação de Segurança\*\***:

- Verificar se senha do admin foi alterada do padrão
- Validar configuração de CORS (apenas domínios autorizados)
- Verificar se rate limiting está ativo
- Validar configuração de logs (não expor dados sensíveis)

7. **\*\*Relatório de Validação\*\***:

- Gerar relatório detalhado com status de cada verificação
- Indicar claramente se o sistema está pronto para produção
- Listar todos os problemas encontrados com prioridade
- Fornecer sugestões de correção para cada problema

O script deve retornar código de saída 0 se todas as validações passarem, ou código diferente de 0 se houver problemas críticos. Usar a classe `CertificateService` existente em `backend/services/certificate-service.js` para

validação de certificados, e o módulo `Database` em  
`backend/config/database.js` para validação de conectividade com MongoDB.

### backend/scripts/backup-producao.sh(NEW)

References:

- scripts/backup.sh

**\*\*Criar script de backup automático para ambiente de produção:\*\***

Script shell para ser executado via cron job diariamente. Implementar:

1. **\*\*Configuração de Variáveis\*\***:

- Diretório de backup: `/backup/brandaocontador-nfe`
- Retenção: manter últimos 30 dias de backups diários, 12 meses de backups mensais
- Timestamp no formato YYYYMMDD\_HHMMSS
- Arquivo de log: `/var/log/backup-nfe.log`

2. **\*\*Backup do MongoDB\*\***:

- Usar `mongodump` com a URI de produção
- Comprimir backup com gzip
- Incluir todas as coleções (usuarios, clientes, produtos, nfes, configuracoes, logs)
- Validar integridade do backup após criação

3. **\*\*Backup de XMLs\*\***:

- Copiar diretório completo `/var/www/brandaocontador-nfe-backend/xmls`

- Incluir subdiretórios (enviadas, falhas)
- Comprimir com tar.gz
- Preservar estrutura de diretórios e permissões

4. **\*\*Backup de Certificados\*\***:

- Copiar diretório `/var/www/brandacontador-nfe-backend/certs``
- Criptografar backup com GPG ou openssl
- Usar senha forte armazenada em local seguro

5. **\*\*Backup de Configurações\*\***:

- Copiar arquivo `.env.production``
- Copiar configurações do nginx
- Copiar configuração do PM2 (`ecosystem.config.js`)

6. **\*\*Upload para Storage Externo\*\***:

- Enviar backup para storage cloud (AWS S3, DigitalOcean Spaces, ou similar)
- Usar credenciais configuradas em variáveis de ambiente
- Validar upload bem-sucedido

7. **\*\*Limpeza de Backups Antigos\*\***:

- Remover backups locais com mais de 30 dias
- Manter backups mensais (primeiro dia de cada mês) por 12 meses
- Remover backups do storage cloud conforme política de retenção

8. **\*\*Notificações\*\***:

- Enviar email de sucesso/falha para administradores
- Incluir tamanho do backup e tempo de execução
- Alertar se espaço em disco estiver baixo (<10GB)

#### 9. **\*\*Logging\*\***:

- Registrar início e fim do backup com timestamp
- Logar tamanho de cada componente do backup
- Registrar erros detalhados se houver falhas
- Rotacionar logs de backup (manter últimos 90 dias)

#### 10. **\*\*Configuração do Cron\*\***:

- Adicionar linha no crontab: `0 2 \* \* \* /var/www/brandaocontador-nfe-backend/scripts/backup-producao.sh`
- Executar às 2h da manhã (horário de menor uso)
- Garantir que script tem permissões de execução (chmod +x)

O script deve ser idempotente e resiliente a falhas, continuando o backup mesmo se uma etapa falhar, mas reportando todos os erros no final.

### backend\scripts\restaurar-backup.sh(NEW)

#### References:

- backend\scripts\backup-producao.sh(NEW)
- backend\scripts\validar-producao.js(NEW)

**\*\*Criar script de restauração de backup para recuperação de desastres:\*\***

Script complementar ao backup para restaurar o sistema em caso de falha.  
Implementar:

#### 1. **\*\*Validação de Pré-requisitos\*\***:



- Verificar se arquivo de backup existe e é válido
- Validar integridade do backup (checksums)
- Confirmar que serviços estão parados antes de restaurar
- Solicitar confirmação do usuário (operação destrutiva)

## 2. **\*\*Restauração do MongoDB\*\***:

- Parar aplicação (PM2) antes de restaurar banco
- Usar `mongorestore` com opção `--drop` para substituir dados existentes
- Validar que todas as coleções foram restauradas
- Verificar contagem de documentos antes/depois

## 3. **\*\*Restauração de XMLs\*\***:

- Descompactar backup de XMLs
- Restaurar para diretório `/var/www/brandaocontador-nfe-backend/xmls`
- Preservar estrutura de diretórios (enviadas, falhas)
- Validar permissões após restauração

## 4. **\*\*Restauração de Certificados\*\***:

- Descriptografar backup de certificados
- Restaurar para diretório `/var/www/brandaocontador-nfe-backend/certs`
- Configurar permissões restritivas (700 para diretório, 600 para arquivos)
- Validar que certificados podem ser carregados

## 5. **\*\*Restauração de Configurações\*\***:

- Restaurar arquivo `.env.production`
- Restaurar configurações do nginx
- Restaurar configuração do PM2
- Recarregar serviços após restauração

6. **\*\*Validação Pós-Restauração\*\***:

- Executar script `validar-producao.js` para verificar integridade
- Testar conectividade com MongoDB
- Validar carregamento de certificado
- Verificar que aplicação inicia corretamente

7. **\*\*Reinício de Serviços\*\***:

- Reiniciar aplicação com PM2
- Recarregar nginx se necessário
- Verificar logs de inicialização
- Confirmar que endpoints estão respondendo

8. **\*\*Teste de Sanidade\*\***:

- Executar teste de health check
- Verificar autenticação
- Testar consulta de dados básicos
- Validar que sistema está operacional

9. **\*\*Logging e Auditoria\*\***:

- Registrar data/hora da restauração
- Logar qual backup foi restaurado
- Registrar usuário que executou a restauração
- Documentar motivo da restauração

10. **\*\*Notificações\*\***:

- Enviar email para administradores sobre restauração
- Incluir detalhes do backup restaurado

- Reportar status de sucesso/falha
- Alertar sobre necessidade de validação manual

O script deve aceitar parâmetros: caminho do backup, data do backup, e opção de restauração parcial (apenas banco, apenas XMLs, etc.). Incluir modo dry-run para simular restauração sem executar.

### GUIA\_PRIMEIRA\_NFE\_PRODUCAO.md(NEW)

References:

- backend\services\nfe-service.js(MODIFY)
- CHECKLIST\_PRODUCAO.md(NEW)

**\*\*Criar guia detalhado para emissão da primeira NFe em produção:\*\***

Documento passo a passo para orientar a equipe na primeira emissão real. Incluir:

1. **\*\*Seção: Preparação\*\***

- Checklist de pré-requisitos (todos os itens de configuração concluídos)
- Validação final do ambiente (executar script `validar-producao.js`)
- Backup completo antes da primeira emissão
- Horário recomendado (horário comercial, com suporte disponível)

2. **\*\*Seção: Dados da Primeira NFe\*\***

- Recomendação: usar operação real mas com valor baixo (ex: R\$ 10,00)
- Dados do destinatário: usar dados reais e válidos
- Produtos: usar produto simples com tributação conhecida
- Natureza da operação: venda simples

- Evitar complexidades na primeira emissão (sem substituição tributária, sem IPI, etc.)

### 3. **\*\*Seção: Processo de Emissão\*\***

- Passo 1: Fazer login no sistema com usuário administrador
- Passo 2: Acessar página de emissão de NFe
- Passo 3: Preencher dados do destinatário (validar CNPJ/CPF)
- Passo 4: Adicionar produto/serviço
- Passo 5: Revisar totais e tributos
- Passo 6: Confirmar emissão
- Passo 7: Aguardar processamento (pode levar 10-30 segundos)

### 4. **\*\*Seção: Validação da Emissão\*\***

- Verificar resposta do sistema (status 100 = autorizado)
- Anotar chave de acesso da NFe (44 dígitos)
- Verificar protocolo de autorização do SEFAZ
- Baixar XML da NFe emitida
- Baixar DANFE (PDF) se disponível

### 5. **\*\*Seção: Validação no Portal do SEFAZ\*\***

- Acessar portal do SEFAZ do estado
- Consultar NFe pela chave de acesso
- Verificar status: deve estar "Autorizada"
- Validar dados da NFe no portal
- Confirmar que XML está disponível para download

### 6. **\*\*Seção: Validação Técnica\*\***

- Verificar logs do backend (PM2 logs)

- Confirmar que XML foi salvo no diretório `xmls/enviadas`
- Validar assinatura digital do XML
- Verificar registro no banco de dados (coleção `nfes`)
- Confirmar que não houve erros nos logs

#### 7. \*\*Seção: O Que Fazer em Caso de Sucesso\*\*

- Documentar chave de acesso e protocolo
- Arquivar XML e DANFE
- Notificar equipe sobre sucesso
- Monitorar sistema nas próximas horas
- Preparar para emissões regulares

#### 8. \*\*Seção: O Que Fazer em Caso de Erro\*\*

- Não entrar em pânico - erros são comuns na primeira emissão
- Anotar código e mensagem de erro completos
- Verificar logs detalhados do backend
- Consultar seção de troubleshooting
- Erros comuns e soluções:
  - \* Erro 403: Certificado não autorizado no SEFAZ
  - \* Erro 218: Certificado expirado ou inválido
  - \* Erro 539: CNPJ do emitente não autorizado
  - \* Erro 280: Certificado não corresponde ao CNPJ

#### 9. \*\*Seção: Cancelamento (Se Necessário)\*\*

- NFe pode ser cancelada em até 24 horas após autorização
- Justificativa mínima de 15 caracteres é obrigatória
- Processo de cancelamento:
  - \* Acessar página de consulta/histórico

- \* Localizar NFe pela chave de acesso
- \* Clicar em "Cancelar NFe"
- \* Informar justificativa detalhada
- \* Confirmar cancelamento
- \* Aguardar processamento
- Validar cancelamento no portal do SEFAZ

#### 10. **\*\*Seção: Próximos Passos\*\***

- Após primeira emissão bem-sucedida, testar outros cenários
- Emitir NFe com múltiplos itens
- Testar diferentes regimes tributários
- Validar cálculos de impostos
- Treinar usuários finais
- Documentar procedimentos operacionais

#### 11. **\*\*Seção: Contatos de Emergência\*\***

- Suporte técnico do sistema
- Suporte da certificadora digital
- Suporte do SEFAZ (telefone e email)
- Contador responsável

#### 12. **\*\*Seção: Checklist Final\*\***

- Lista de verificação com todos os passos
- Espaço para anotações e observações
- Assinatura do responsável pela emissão
- Data e hora da primeira emissão

Formatar com destaque visual para avisos importantes e passos críticos. Incluir screenshots ou diagramas se possível.

### TROUBLESHOOTING\_PRODUCAO.md(NEW)

References:

- backend\services\nfe-service.js(MODIFY)
- backend\services\sefaz-client.js
- backend\ws\_urls\_uf.js

**\*\*Criar guia completo de troubleshooting para problemas em produção:\*\***

Documento de referência rápida para resolver problemas comuns. Organizar por categoria:

#### 1. **\*\*Problemas de Certificado Digital\*\***

- **\*\*Erro: "Certificado não carregado"\*\***
  - \* Causa: CERT\_PATH incorreto ou arquivo não existe
  - \* Solução: Verificar caminho no .env.production, confirmar que arquivo existe com `ls -la``
- **\*\*Erro: "Senha do certificado incorreta"\*\***
  - \* Causa: CERT\_PASS incorreto
  - \* Solução: Validar senha com a certificadora, testar localmente
- **\*\*Erro 218: "Certificado expirado"\*\***
  - \* Causa: Certificado vencido
  - \* Solução: Renovar certificado A1, atualizar no servidor
- **\*\*Erro 280: "Certificado não corresponde ao CNPJ"\*\***
  - \* Causa: Certificado de outra empresa

\* Solução: Usar certificado correto da empresa emitente

## 2. \*\*Problemas de Conectividade\*\*

- \*\*Erro: "ECONNREFUSED MongoDB"\*\*

\* Causa: MongoDB não está rodando ou URI incorreta

\* Solução: Verificar status do MongoDB, validar MONGODB\_URI, checar whitelist de IPs

- \*\*Erro: "Timeout ao conectar com SEFAZ"\*\*

\* Causa: Firewall bloqueando, SEFAZ fora do ar, timeout muito baixo

\* Solução: Verificar firewall do servidor, aumentar TIMEOUT, consultar status do SEFAZ

- \*\*Erro 403: "Forbidden"\*\*

\* Causa: Certificado não autorizado no SEFAZ

\* Solução: Cadastrar certificado no portal do SEFAZ, verificar autorização para NFe

## 3. \*\*Problemas de Configuração\*\*

- \*\*Erro: "CNPJ\_EMITENTE não configurado"\*\*

\* Causa: Variável vazia no .env

\* Solução: Configurar CNPJ no .env.production, reiniciar aplicação

- \*\*Erro: "UF não encontrada"\*\*

\* Causa: UF inválida ou não suportada

\* Solução: Verificar código da UF no arquivo `ws\_urls\_uf.js`, usar código correto

- \*\*Erro: "Ambiente inválido"\*\*

\* Causa: AMBIENTE diferente de 1 ou 2

\* Solução: Configurar AMBIENTE=1 para produção ou AMBIENTE=2 para homologação

## 4. \*\*Problemas de Emissão\*\*



- **\*\*Erro 539: "CNPJ não autorizado para emissão"\*\***
  - \* Causa: CNPJ não cadastrado no SEFAZ para NFe
  - \* Solução: Solicitar credenciamento no portal do SEFAZ
- **\*\*Erro 225: "Rejeição: Falha no schema XML"\*\***
  - \* Causa: XML mal formado ou campos obrigatórios faltando
  - \* Solução: Validar dados de entrada, verificar logs do método `gerarXmlNfe``
- **\*\*Erro 204: "Duplicidade de NFe"\*\***
  - \* Causa: Tentativa de emitir NFe com número já usado
  - \* Solução: Incrementar numeração, verificar último número usado

#### 5. **\*\*Problemas de Performance\*\***

- **\*\*Sistema lento para emitir NFe\*\***
  - \* Causa: Timeout baixo, SEFAZ lento, servidor sobrecarregado
  - \* Solução: Aumentar TIMEOUT, verificar recursos do servidor (CPU, memória)
- **\*\*Banco de dados lento\*\***
  - \* Causa: Queries não otimizadas, índices faltando
  - \* Solução: Criar índices nas coleções principais, otimizar queries

#### 6. **\*\*Problemas de Logs e Monitoramento\*\***

- **\*\*Logs não estão sendo gerados\*\***
  - \* Causa: Diretório de logs sem permissão de escrita
  - \* Solução: Criar diretório, configurar permissões (`chmod 755`)
- **\*\*Logs muito grandes\*\***
  - \* Causa: `LOG_LEVEL=debug` em produção
  - \* Solução: Configurar `LOG_LEVEL=info` ou `warn`, implementar rotação de logs

#### 7. **\*\*Problemas de Segurança\*\***

- **\*\*Erro: "Token JWT inválido"\*\***

- \* Causa: Token expirado ou JWT\_SECRET alterado
- \* Solução: Fazer novo login, verificar JWT\_SECRET não foi alterado
- \*\*Erro: "CORS bloqueado"\*\*
- \* Causa: Origem não autorizada em CORS\_ORIGINS
- \* Solução: Adicionar domínio à lista de origens permitidas

## 8. \*\*Comandos Úteis para Diagnóstico\*\*

```
` `` bash
```

```
# Verificar status da aplicação
```

```
pm2 status
```

```
pm2 logs --lines 100
```

```
# Verificar conectividade MongoDB
```

```
mongosh "$MONGODB_URI" --eval "db.adminCommand('ping')"
```

```
# Verificar certificado
```

```
openssl pkcs12 -info -in certificado.pfx -noout
```

```
# Verificar espaço em disco
```

```
df -h
```

```
# Verificar uso de memória
```

```
free -h
```

```
# Verificar logs do nginx
```

```
tail -f /var/log/nginx/error.log
```

```
# Testar endpoint de health
```

curl https://api.brandaocontador.com.br/health

` `` `

#### 9. **Procedimentos de Recuperação**

##### - **Aplicação não inicia**

- \* Verificar logs: `pm2 logs`
- \* Validar .env: `cat .env.production`
- \* Testar manualmente: `node app-real.js`

##### - **Banco de dados corrompido**

- \* Parar aplicação: `pm2 stop all`
- \* Restaurar backup: `./scripts/restaurar-backup.sh`
- \* Validar integridade: `./scripts/validar-producao.js`

##### - **Certificado comprometido**

- \* Revogar certificado imediatamente
- \* Obter novo certificado
- \* Atualizar no servidor
- \* Notificar SEFAZ se necessário

#### 10. **Escalação de Problemas**

##### - **Nível 1**: Problemas de configuração e operação

- \* Resolver com documentação e scripts

##### - **Nível 2**: Problemas técnicos complexos

- \* Contatar suporte técnico do sistema

##### - **Nível 3**: Problemas com SEFAZ ou certificadora

- \* Contatar suporte oficial do SEFAZ ou certificadora

##### - **Nível 4**: Problemas críticos de segurança

- \* Escalar imediatamente, parar sistema se necessário

Incluir tabela de códigos de erro do SEFAZ mais comuns com descrição e solução.  
Adicionar links para documentação oficial do SEFAZ e manuais técnicos.

### backend\models\Configuracao.js(MODIFY)

References:

- backend\services\Nfe-service.js(MODIFY)
- backend\config\database.js

**\*\*Validar e popular configurações da empresa no banco de dados:\*\***

O model `Configuracao` já está corretamente implementado com todos os campos necessários. Antes de ativar produção, é necessário:

1. **\*\*Criar/Atualizar Documento de Configuração\*\***:

- Acessar o banco de dados MongoDB de produção
- Criar ou atualizar documento com `chave: 'padrao'` na coleção `configuracoes`
- Preencher todos os campos obrigatórios da seção `empresa`:
  - \* `razaoSocial` : Razão social completa da empresa
  - \* `nomeFantasia` : Nome fantasia
  - \* `cnpj` : CNPJ formatado ou apenas números
  - \* `inscricaoEstadual` : IE do estado
  - \* `inscricaoMunicipal` : IM se aplicável
  - \* `email` : Email corporativo
  - \* `telefone` : Telefone de contato
  - \* `formaTributacao` : Regime tributário (simples\_nacional, lucro\_presumido, lucro\_real, mei)

\* `endereco` : Objeto completo com CEP, logradouro, número, bairro, município, UF

## 2. **\*\*Configurar Seção NFe\*\***:

- `ambiente` : 'producao' (importante: string, não número)
- `serie` : Série da NFe (geralmente '1')
- `numeracaoInicial` : Número inicial da numeração (geralmente 1)
- `certificadoDigital.arquivo` : Caminho absoluto do certificado no servidor
- `certificadoDigital.senha` : Senha do certificado (criptografada se possível)
- `certificadoDigital.validade` : Data de validade do certificado
- `certificadoDigital.status` : 'ativo'

## 3. **\*\*Configurar Notificações\*\***:

- Definir quais notificações devem ser enviadas (emailNFeEmitida, emailNFeCancelada, etc.)
- Configurar servidor de email se notificações estiverem habilitadas
- Validar que email de envio está configurado corretamente

## 4. **\*\*Validação dos Dados\*\***:

- O método `carregarCertificadoSistema` em `nfe-service.js` (linhas 52-68) já implementa a lógica de carregar certificado do banco
- Prioridade: banco de dados > variáveis de ambiente
- Garantir que dados no banco estão corretos e completos

## 5. **\*\*Script de População\*\***:

- Criar script auxiliar para popular configurações iniciais
- Validar dados antes de inserir no banco
- Permitir atualização via interface administrativa
- Implementar versionamento de configurações para auditoria

**\*\*Importante\*\***: O schema já define valores padrão para campos opcionais, mas campos críticos como CNPJ, razão social e endereço devem ser preenchidos manualmente. O sistema não funcionará corretamente em produção sem essas configurações.

### CRONOGRAMA\_ATIVACAO\_PRODUCAO.md(NEW)

References:

- CHECKLIST\_PRODUCAO.md(NEW)
- GUIA\_PRIMEIRA\_NFE\_PRODUCAO.md(NEW)

**\*\*Criar cronograma detalhado para ativação do ambiente de produção:\*\***

Documento de planejamento com timeline realista para ativação. Estruturar por fases:

1. **\*\*Fase 0: Preparação (Dia 1)\*\***

- Manhã:

- \* Reunião de kickoff com equipe técnica e stakeholders
- \* Revisão completa do checklist de produção
- \* Definição de responsáveis por cada tarefa
- \* Alinhamento de expectativas e riscos

- Tarde:

- \* Solicitar certificado digital A1 junto à certificadora
- \* Coletar todos os dados da empresa (CNPJ, IE, endereço completo)
- \* Provisionar MongoDB de produção (MongoDB Atlas ou local)
- \* Configurar whitelist de IPs no MongoDB

- Entregáveis:

- \* Certificado digital solicitado (prazo: 1-3 dias úteis)
- \* MongoDB provisionado e acessível
- \* Dados da empresa coletados e validados

## 2. \*\*Fase 1: Configuração de Infraestrutura (Dias 2-3)\*\*

- Dia 2 - Manhã:

- \* Receber certificado digital da certificadora
- \* Validar certificado localmente (senha, validade, formato)
- \* Fazer upload seguro do certificado para o servidor
- \* Configurar permissões restritivas no diretório de certificados

- Dia 2 - Tarde:

- \* Configurar todas as variáveis de ambiente no ``.env.production``
- \* Gerar `JWT_SECRET` forte e único
- \* Configurar string de conexão do MongoDB
- \* Validar conectividade do servidor com MongoDB

- Dia 3 - Manhã:

- \* Popular configurações da empresa no banco de dados (model ``Configuracao``)
- \* Configurar dados fiscais (regime tributário, série NFe, numeração)
- \* Criar usuário administrador com senha forte
- \* Executar script ``validar-producao.js`` para verificar configurações

- Dia 3 - Tarde:

- \* Configurar backup automático (script e cron job)
- \* Testar processo de backup e restauração
- \* Configurar monitoramento básico (logs, alertas)
- \* Documentar todas as configurações realizadas

- Entregáveis:

- \* Certificado instalado e validado no servidor
- \* Todas as variáveis de ambiente configuradas
- \* Banco de dados populado com configurações
- \* Backup automático funcionando

### 3. \*\*Fase 2: Testes em Homologação com Certificado Real (Dias 4-5)\*\*

#### - Dia 4 - Manhã:

- \* Configurar ambiente de homologação com certificado real
- \* Alterar AMBIENTE=2 no ` .env.homologacao `
- \* Reiniciar aplicação em modo homologação
- \* Verificar logs de inicialização

#### - Dia 4 - Tarde:

- \* Teste 1: Emitir NFe simples (1 item, sem complexidades)
- \* Teste 2: Emitir NFe com múltiplos itens
- \* Teste 3: Consultar NFe emitida
- \* Teste 4: Cancelar NFe de teste

#### - Dia 5 - Manhã:

- \* Teste 5: Inutilizar numeração
- \* Teste 6: Emitir NFe com diferentes regimes tributários
- \* Teste 7: Validar XMLs gerados (estrutura, assinatura)
- \* Teste 8: Verificar persistência no banco de dados

#### - Dia 5 - Tarde:

- \* Validar todas as NFes no portal do SEFAZ
- \* Analisar logs de todas as operações
- \* Documentar problemas encontrados e soluções
- \* Aprovar ou reprovar ambiente para produção

#### - Entregáveis:

- \* Relatório de testes com todos os cenários validados



- \* Lista de problemas encontrados e resolvidos
- \* Aprovação formal para migração para produção

#### 4. \*\*Fase 3: Preparação Final para Produção (Dia 6)\*\*

##### - Manhã:

- \* Backup completo do ambiente atual
- \* Revisão final de todas as configurações
- \* Executar script `validar-producao.js` novamente
- \* Preparar plano de rollback detalhado

##### - Tarde:

- \* Alterar AMBIENTE=1 no ` .env.production`
- \* Configurar NODE\_ENV=production
- \* Reiniciar aplicação com configurações de produção
- \* Verificar logs de inicialização (sem erros)
- \* Testar endpoint de health check
- \* Validar que sistema está em modo produção (SIMULATION\_MODE=false)

##### - Entregáveis:

- \* Sistema configurado para produção
- \* Backup pré-produção realizado
- \* Plano de rollback documentado

#### 5. \*\*Fase 4: Primeira Emissão Real (Dia 7)\*\*

##### - Manhã:

- \* Reunião de go-live com equipe técnica
- \* Revisão final do guia de primeira emissão
- \* Preparar dados para primeira NFe (valor baixo, operação simples)
- \* Garantir que suporte técnico está disponível

##### - Meio-dia:

\* \*\*EMISSÃO DA PRIMEIRA NFE REAL\*\*

- \* Monitorar logs em tempo real
- \* Validar resposta do SEFAZ (status 100)
- \* Anotar chave de acesso e protocolo
- \* Baixar XML e DANFE

- Tarde:

- \* Validar NFe no portal do SEFAZ
- \* Verificar registro no banco de dados
- \* Analisar logs completos da operação
- \* Documentar resultado (sucesso ou problemas)
- \* Se sucesso: comemorar e preparar para operação regular
- \* Se falha: executar troubleshooting e tentar novamente

- Entregáveis:

- \* Primeira NFe real emitida e autorizada
- \* Documentação completa da emissão
- \* Sistema validado em produção

6. \*\*Fase 5: Estabilização e Monitoramento (Dias 8-14)\*\*

- Dias 8-10:

- \* Emitir NFes reais com diferentes cenários
- \* Monitorar performance e estabilidade
- \* Ajustar configurações se necessário
- \* Treinar usuários finais

- Dias 11-14:

- \* Operação assistida (suporte técnico acompanhando)
- \* Documentar procedimentos operacionais
- \* Criar FAQs para usuários
- \* Implementar melhorias identificadas

- Entregáveis:

- \* Sistema estável e operacional
- \* Usuários treinados
- \* Documentação operacional completa

7. **\*\*Marcos e Decisões Críticas\*\***:

- **\*\*Go/No-Go Fase 2 → Fase 3\*\***: Todos os testes em homologação devem passar
- **\*\*Go/No-Go Fase 3 → Fase 4\*\***: Script de validação deve retornar 100% OK
- **\*\*Go/No-Go Fase 4 → Fase 5\*\***: Primeira NFe deve ser autorizada com sucesso

8. **\*\*Riscos e Mitigações\*\***:

- **\*\*Risco\*\***: Certificado não chega no prazo
  - \* Mitigação: Solicitar com antecedência, ter certificadora backup
- **\*\*Risco\*\***: Testes em homologação falham
  - \* Mitigação: Alocar tempo extra para correções (buffer de 2 dias)
- **\*\*Risco\*\***: Primeira emissão real falha
  - \* Mitigação: Ter plano de rollback pronto, suporte técnico disponível
- **\*\*Risco\*\***: SEFAZ fora do ar no dia da ativação
  - \* Mitigação: Consultar calendário de manutenções, ter data alternativa

9. **\*\*Recursos Necessários\*\***:

- Equipe técnica: 1-2 desenvolvedores full-time
- Suporte: Certificadora, SEFAZ, contador
- Infraestrutura: Servidor DigitalOcean, MongoDB Atlas
- Ferramentas: Acesso SSH, MongoDB Compass, Postman
- Tempo total estimado: 7-14 dias úteis

10. **\*\*Critérios de Sucesso\*\***:

- ☒ Certificado digital instalado e validado
- ☒ Todas as configurações corretas e validadas
- ☒ Testes em homologação 100% aprovados
- ☒ Primeira NFe real autorizada pelo SEFAZ
- ☒ Sistema estável por 7 dias consecutivos
- ☒ Usuários treinados e operando autonomamente

Incluir diagrama de Gantt visual e tabela de responsabilidades (RACI matrix) para cada fase.