Documentação Formal do Sistema - SIGVoIP

Sumário

[1. Resumo Executivo 1](#_Toc202345768)

[2. Visão Geral do Sistema 2](#_Toc202345769)

[3. Contexto dos Sistemas Envolvidos 2](#_Toc202345770)

[4. Arquitetura de Alto Nível 2](#_Toc202345771)

[5. Modelo de Domínio 3](#_Toc202345772)

[6. Diagrama de Classes 4](#_Toc202345773)

[7. Diagrama Entidade-Relacionamento (DER) 4](#_Toc202345774)

[8. Casos de Uso 5](#_Toc202345775)

[9. Requisitos Funcionais 6](#_Toc202345776)

[10. Requisitos Não Funcionais 7](#_Toc202345777)

[11. Estratégia de Integração entre Sistemas 8](#_Toc202345778)

[12. Regras de Negócio 9](#_Toc202345779)

[13. Plano de Testes 10](#_Toc202345780)

[1. Tipos de Testes: 10](#_Toc202345781)

[2. Ambiente de Testes: 10](#_Toc202345782)

[14. Roadmap de Etapas Futuras 11](#_Toc202345783)

[15. Glossário de Termos 11](#_Toc202345784)

[16. Histórico de Versões da Documentação 13](#_Toc202345785)

# ****Resumo Executivo****

O SIGVoIP é um projeto estratégico focado na automação da integração de dados entre os sistemas DocSales (CRM), Omie (ERP) e as plataformas NextBilling/NextRouter (VoIP). Atualmente, a ausência de um elo unificado entre esses sistemas resulta em processos manuais propensos a erros, retrabalho e potenciais perdas financeiras.

Este sistema visa centralizar e otimizar o fluxo de trabalho pós-venda, garantindo a governança e a rastreabilidade dos dados de clientes e serviços. A Fase 1 prioriza a automatização da ativação de serviços e faturamento para propostas aprovadas, reduzindo significativamente o tempo de ativação e eliminando inconsistências. O SIGVoIP proporcionará um painel administrativo para monitoramento, controle e reprocessamento de integrações, fortalecendo a segurança e a eficiência operacional da empresa.

# ****Visão Geral do Sistema****

O SIGVoIP (Sistema de Integração e Gestão para Telefonia VoIP) é uma solução estratégica desenvolvida para automatizar e otimizar a interação entre os sistemas cruciais da empresa: DocSales (CRM), Omie (ERP) e as plataformas NextBilling/NextRouter (sistemas VoIP).

Seu propósito principal é eliminar gargalos operacionais e inconsistências de dados resultantes de processos manuais de integração. Ao centralizar o fluxo de informações, o SIGVoIP visa reduzir erros humanos, acelerar o onboarding de novos clientes e serviços, e mitigar perdas financeiras associadas a falhas na sincronização de dados. O sistema atuará como uma fonte unificada de verdade para os dados de clientes e serviços, garantindo governança e rastreabilidade em todo o ciclo de vida da integração.

A Fase 1 do projeto foca na automação completa do fluxo de propostas aprovadas originadas no DocSales até o provisionamento de serviços no Omie e Next, garantindo um processo de pós-venda eficiente e sem atritos.

# ****Contexto dos Sistemas Envolvidos****

O SIGVoIP opera como o elo central entre três ecossistemas de software distintos, cada um com suas responsabilidades e APIs:

* **DocSales (CRM):** Sistema de gestão de relacionamento com o cliente e propostas comerciais. É a **fonte primária** de informações sobre novas propostas e contratos aprovados que desencadeiam o processo de integração. O SIGVoIP consultará a API DocSales para identificar propostas no status "approved".
* **Omie (ERP):** Sistema de gestão empresarial. Após a aprovação de uma proposta, o SIGVoIP será responsável por criar ou atualizar os registros de clientes na Omie e, crucialmente, configurar as contas recorrentes e lançamentos financeiros relacionados aos serviços contratados via API.
* **NextBilling / NextRouter (Plataformas VoIP):** Sistemas de provisionamento e gestão de serviços de telefonia VoIP. O SIGVoIP orquestrará o cadastro/atualização de assinantes e a configuração de seus respectivos serviços (ramais, DIDs, planos) nessas plataformas. Inclui a lógica para escolher o servidor Next ideal com base na capacidade disponível.

O SIGVoIP atua como uma camada de orquestração e tradução entre esses sistemas, garantindo que as informações fluam de forma correta e consistente, aplicando regras de negócio e validações específicas a cada integração.

(Nota: Um diagrama visual de contexto de sistemas (docs/arquitetura/visual/17\_contexto\_sistemas.png) deve ser inserido aqui para ilustrar graficamente as interações. Você pode referenciá-lo com: "Para uma representação visual do contexto de sistemas, consulte a Figura 3.1 abaixo.")

# ****Arquitetura de Alto Nível****

O sistema SIGVoIP é construído com uma arquitetura em camadas, fortemente influenciada pelos princípios da Clean Architecture e da Arquitetura Hexagonal (Ports and Adapters). Esta abordagem visa garantir alta modularidade, testabilidade, manutenibilidade e independência da lógica de negócio em relação a detalhes de infraestrutura ou tecnologias externas.

As principais camadas são:

* **Domínio:** O coração do sistema. Contém toda a lógica de negócio essencial, entidades (ClienteSIG, UsuarioSIG, ServidorNext), Value Objects (DocumentoFiscal, Endereco, Contato), e interfaces de repositório. É completamente agnóstica a frameworks e bancos de dados.
* **Aplicação:** Orquestra os casos de uso do sistema, definindo as operações que podem ser realizadas. Depende apenas da camada de Domínio, utilizando suas interfaces.
* **Infraestrutura:** Implementa as interfaces definidas nas camadas de Domínio e Aplicação. É onde se encontram os adaptadores para tecnologias externas, como a persistência de dados (Entity Framework Core com MySQL) e os clientes das APIs externas (DocSales, Omie, Next).
* **Apresentação (API RESTful):** A camada mais externa, responsável pela interface do usuário (ou de outros sistemas). Atua como um adaptador, recebendo requisições HTTP e traduzindo-as para os casos de uso da camada de Aplicação. Inclui Controladores, DTOs e gerenciamento de autenticação/autorização (ASP.NET Core Identity + JWT).

A Regra de Dependência é estritamente aplicada: as dependências de código sempre apontam para as camadas mais internas, garantindo que o Domínio permaneça o mais isolado e estável possível.

(Nota: Um diagrama visual da arquitetura de alto nível (docs/arquitetura/visual/08\_diagrama\_arquitetura.png ou docs/arquitetura/visual/arquitetura-alto-nivel.png) deve ser inserido aqui para ilustrar a estrutura de camadas. Você pode referenciá-lo com: "A Figura 4.1 ilustra a arquitetura de alto nível do SIGVoIP.")

# ****Modelo de Domínio****

O Modelo de Domínio do SIGVoIP é o conjunto de conceitos e regras de negócio centrais, independentes de tecnologias de implementação. Ele é construído em torno de Agregados e Value Objects, com foco na Linguagem Ubíqua do negócio.

As principais entidades e Value Objects modelados incluem:

* **Agregados e Entidades Raiz:**
  + **ClienteSIG**: A entidade central e mais importante do sistema, representando o cliente de forma unificada e agnóstica às representações em sistemas externos. É a Aggregate Root principal. Gerencia contatos, endereços e integrações com sistemas externos através de objetos aninhados.
  + **UsuarioSIG**: Representa um usuário interno do sistema SIGVoIP, com credenciais e perfis de acesso.
  + **ServidorNext**: Representa um servidor da plataforma Next, com informações de capacidade e status.
* **Value Objects:**
  + **DocumentoFiscal**: Encapsula CPF ou CNPJ, com validação de formato e tipo.
  + **CEP**: Código de Endereçamento Postal, com validação de formato.
  + **Endereco**: Endereço completo, incluindo rua, número, complemento, bairro, cidade, estado e país.
  + **Email**: Endereço de e-mail, com validação de formato.
  + **Telefone**: Número de telefone, DDD e ramal opcional, com validação de formato.
  + **ContatoTelefonico**: Agrega um Telefone com informações de uso (pessoal/comercial) e tipo (fixo/celular).
  + **PessoaContato**: Representa uma pessoa de contato específica para um Cliente Empresa, com nome, cargo e seu próprio conjunto de canais de comunicação.
  + **ClienteNextIntegracao**: VO para armazenar identificadores e dados de mapeamento de clientes no sistema Next.
  + **ClienteOmieIntegracao**: VO para armazenar identificadores e dados de mapeamento de clientes no sistema Omie.
  + **ServicoContratado**: Nova entidade (ou VO complexo, a depender do detalhamento) que encapsula os detalhes de um serviço ou contrato específico e suas IDs de mapeamento para DocSales, Omie e Next, permitindo múltiplos serviços para um único ClienteSIG.

As regras de negócio associadas a esses elementos de domínio são encapsuladas dentro deles, garantindo a consistência e a validade dos dados em qualquer parte do sistema.

(Nota: Um diagrama visual do modelo de domínio (docs/arquitetura/visual/18\_modelo\_dominio.png) deve ser inserido aqui para ilustrar as entidades e VOs. Você pode referenciá-lo com: "A Figura 5.1 apresenta o Modelo de Domínio do SIGVoIP.")

# ****Diagrama de Classes****

(Esta seção requer a inserção do diagrama visual. A descrição abaixo detalha o conteúdo esperado.)

O Diagrama de Classes UML (Unified Modeling Language) apresentará uma visão detalhada da estrutura estática do sistema SIGVoIP, focando nas classes do domínio e suas principais interações.

O diagrama (docs/arquitetura/visual/05\_diagrama\_classes.png) ilustrará:

* **Classes/Entidades Principais:** Aggregate Roots como ClienteSIG, e Value Objects como DocumentoFiscal, Endereco, Email, Telefone, além de entidades como UsuarioSIG, ServidorNext e ServicoContratado.
* **Relacionamentos:** Associações (um-para-um, um-para-muitos), agregação, composição, herança e dependência entre as classes.
* **Atributos e Métodos Relevantes:** Visibilidade, tipo de dado e comportamento essencial de cada classe.

O diagrama também representará as interfaces de repositório (e.g., IClienteSIGRepository) e as interfaces para serviços de integração (e.g., IDocSalesIntegrationService), mostrando como as camadas de Domínio e Aplicação dependem de abstrações, e como a camada de Infraestrutura as implementa.

(Inserir Diagrama de Classes UML aqui: docs/arquitetura/visual/05\_diagrama\_classes.png)

# ****Diagrama Entidade-Relacionamento (DER)****

(Esta seção requer a inserção do diagrama visual. A descrição abaixo detalha o conteúdo esperado.)

O Diagrama Entidade-Relacionamento (DER) representará a estrutura do banco de dados relacional (MySQL) que suportará a persistência do sistema SIGVoIP. Ele mostrará as entidades (tabelas), seus atributos (colunas) e os relacionamentos entre elas, incluindo chaves primárias e estrangeiras.

O diagrama (docs/arquitetura/visual/06\_DER.png) exibirá:

* **Entidades Chave:**
  + Clientes: Tabela principal para a entidade ClienteSIG.
  + Usuarios: Tabela para a entidade UsuarioSIG.
  + ServidoresNext: Tabela para a entidade ServidorNext.
  + ServicosContratados: Tabela para a entidade ServicoContratado, detalhando cada serviço/contrato e suas associações com IDs externas.
  + Tabelas auxiliares para dados de VOs persistíveis (se houver a decisão de persistir VOs complexos como entidades independentes ou através de tabelas separadas para coleções), como ContatosClientes ou EmailsClientes.
  + Tabelas de Log de Integração.
* **Atributos:** Nomes das colunas, tipos de dados e restrições de nulidade.
* **Relacionamentos:** Chaves primárias (PK), chaves estrangeiras (FK) e a cardinalidade dos relacionamentos (um-para-um, um-para-muitos, muitos-para-muitos).

A modelagem seguirá a tradução das entidades e Value Objects do domínio para um esquema relacional otimizado para MySQL, com foco em normalização e performance.

(Inserir Diagrama Entidade-Relacionamento (DER) aqui: docs/arquitetura/visual/06\_DER.png)

# ****Casos de Uso****

Os Casos de Uso descrevem as interações funcionais entre os atores (usuários ou sistemas externos) e o sistema SIGVoIP, definindo as funcionalidades que o sistema deve oferecer do ponto de vista do usuário.

(Nota: Um diagrama visual de Casos de Uso UML (docs/planejamento/visual/03\_casos\_de\_uso.png - assumindo que será gerado aqui) deve ser inserido para uma visão gráfica. Você pode referenciá-lo com: "A Figura 8.1 apresenta o diagrama de casos de uso do SIGVoIP.")

Atores Principais:

* **Usuário Administrador SIGVoIP:** Gerencia configurações, visualiza logs, reprocessa integrações, gerencia usuários.
* **Usuário Financeiro SIGVoIP:** Visualiza status de integração, acessa relatórios financeiros (funcionalidade futura).
* **Sistema DocSales:** Ator externo que dispara eventos ou fornece dados de propostas.
* **Sistema Omie:** Ator externo que recebe dados de clientes e contas a receber.
* **Sistema NextBilling/NextRouter:** Ator externo que recebe dados de assinantes e configurações de serviços.

Principais Casos de Uso (Fase 1):

* **UC1: Importar Proposta Aprovada do DocSales**
  + **Descrição:** O sistema verifica periodicamente novas propostas "aprovadas" no DocSales para iniciar o processo de integração.
  + **Ator Primário:** Sistema DocSales (disparador) / SIGVoIP (Executor).
  + **Fluxo Principal:** SIGVoIP consulta API DocSales -> Identifica novas propostas -> Recebe payload -> Inicia processo de integração.
* **UC2: Cadastrar/Atualizar Cliente no Omie**
  + **Descrição:** Com base nos dados da proposta, o SIGVoIP cria ou atualiza o cliente correspondente no Omie.
  + **Ator Primário:** SIGVoIP.
  + **Fluxo Principal:** SIGVoIP extrai dados do ClienteSIG -> Mapeia para ClienteOmie IntegMod -> Chama API Omie para cadastro/atualização -> Armazena IdOmie no ClienteSIG.
* **UC3: Gerar Conta a Receber Recorrente no Omie**
  + **Descrição:** O SIGVoIP configura as contas a receber para os serviços contratados no Omie, com base na recorrência da proposta.
  + **Ator Primário:** SIGVoIP.
  + **Fluxo Principal:** SIGVoIP extrai dados das parcelas e serviços -> Mapeia para ContaReceberOmie IntegMod -> Chama API Omie para criação da conta recorrente.
* **UC4: Cadastrar/Atualizar Assinante no Next**
  + **Descrição:** O SIGVoIP cria ou atualiza o assinante na plataforma Next com base nos dados do cliente e serviços.
  + **Ator Primário:** SIGVoIP.
  + **Fluxo Principal:** SIGVoIP extrai dados do ClienteSIG e serviços -> Mapeia para AssinanteNext IntegMod -> Seleciona servidor Next (capacidade) -> Chama API Next para cadastro/atualização -> Armazena IdNext no ClienteSIG (ou ServicoContratado).
* **UC5: Gerenciar Erros de Integração**
  + **Descrição:** O sistema registra e permite o reprocessamento de propostas que falharam na integração.
  + **Ator Primário:** Usuário Administrador SIGVoIP.
  + **Fluxo Principal:** Erro detectado -> Log registrado -> Proposta marcada para reprocessamento -> Administrador visualiza erros e aciona reprocessamento.
* **UC6: Gerenciar Usuários do Sistema**
  + **Descrição:** Permite que o Administrador crie, edite e remova usuários internos do SIGVoIP.
  + **Ator Primário:** Usuário Administrador SIGVoIP.
  + **Fluxo Principal:** Administrador acessa painel -> Cria novo usuário -> Edita/inativa usuário existente.

(Nota: Para detalhes e fluxos completos de cada Caso de Uso, consulte o docs/planejamento/03\_casos\_de\_uso.md.)

# ****Requisitos Funcionais****

Os Requisitos Funcionais definem as capacidades que o sistema SIGVoIP deve possuir para atender às necessidades do negócio.

* **RF001 - Importação de Propostas DocSales:** O sistema deve ser capaz de consultar a API DocSales para identificar propostas com status "approved".
* **RF002 - Seleção de Propostas para Integração:** O sistema deve permitir a seleção manual de propostas aprovadas a serem processadas para integração.
* **RF003 - Cadastro/Atualização de Cliente Omie:** O sistema deve ser capaz de cadastrar um novo cliente na Omie ou atualizar um cliente existente, utilizando os dados do ClienteSIG.
* **RF004 - Criação de Conta a Receber Recorrente Omie:** O sistema deve ser capaz de criar contas a receber recorrentes na Omie com base nas parcelas e vigência dos serviços da proposta.
* **RF005 - Cadastro/Atualização de Assinante Next:** O sistema deve ser capaz de cadastrar um novo assinante ou atualizar um existente na plataforma Next.
* **RF006 - Alocação Automática de Servidor Next:** O sistema deve ser capaz de selecionar o servidor Next ideal para um novo assinante, considerando sua capacidade e configurações.
* **RF007 - Mapeamento de IDs Externas:** O sistema deve armazenar e associar os IDs de clientes gerados no Omie e Next ao respectivo ClienteSIG (ou ServicoContratado).
* **RF008 - Registro de Logs de Integração:** O sistema deve registrar detalhadamente o status (sucesso/falha) de cada etapa do processo de integração.
* **RF009 - Reprocessamento de Propostas:** O sistema deve permitir que um usuário administrador reprocesse propostas que apresentaram erro em integrações anteriores.
* **RF010 - Autenticação de Usuários:** O sistema deve autenticar usuários internos (Administrador, Financeiro) utilizando usuário e senha (ASP.NET Core Identity + JWT).
* **RF011 - Autorização por Perfil:** O sistema deve controlar o acesso a funcionalidades e dados com base nos perfis de usuário (Administrador, Financeiro).
* **RF012 - Painel Administrativo de Integração:** O sistema deve prover um painel para visualizar o status das integrações, logs de erro e opções de reprocessamento.
* **RF013 - Validação de Dados de Entrada:** O sistema deve validar os dados recebidos das APIs externas antes do processamento e antes de enviá-los para outras APIs.
* **RF014 - Notificação de Erros Críticos:** O sistema deve notificar administradores sobre falhas críticas de integração.

(Nota: Para detalhes e critérios de aceitação completos para cada requisito, consulte o docs/planejamento/02\_documento\_requisitos.md.)

# ****Requisitos Não Funcionais****

Os Requisitos Não Funcionais definem os critérios de qualidade e as restrições sob as quais o sistema deve operar.

* **RNF001 - Desempenho:**
  + **RNF001.1:** O processo de importação de propostas do DocSales deve ser concluído em até X minutos para Y propostas (a ser definido).
  + **RNF001.2:** A integração completa de uma única proposta com Omie e Next deve ocorrer em menos de 5 segundos.
* **RNF002 - Segurança:**
  + **RNF002.1:** O sistema deve garantir a confidencialidade e integridade dos dados, utilizando criptografia para senhas e dados sensíveis em trânsito e em repouso.
  + **RNF002.2:** O sistema deve estar em conformidade com as diretrizes de segurança da OWASP (Open Web Application Security Project).
  + **RNF002.3:** A autenticação de usuários deve ser robusta, incluindo proteção contra ataques de força bruta e uso de tokens JWT com expiração.
* **RNF003 - Disponibilidade:**
  + **RNF003.1:** O sistema deve operar com uma disponibilidade de 99.5% em horário comercial (8h-18h, seg-sex).
  + **RNF003.2:** A janela de manutenção planejada não deve exceder 2 horas por mês.
* **RNF004 - Escalabilidade:**
  + **RNF004.1:** A arquitetura do sistema deve permitir o escalonamento horizontal para suportar um aumento de 50% no volume de propostas e usuários em 12 meses.
  + **RNF004.2:** A camada de persistência deve ser capaz de lidar com um crescimento de dados de até 100GB/ano.
* **RNF005 - Manutenibilidade:**
  + **RNF005.1:** O código-fonte deve aderir aos princípios de Clean Code, SOLID e padrões de projeto, conforme detalhado nas Diretrizes de Desenvolvimento.
  + **RNF005.2:** O sistema deve possuir uma cobertura de testes unitários superior a 80% na camada de Domínio e Aplicação.
  + **RNF005.3:** A documentação técnica (Wiki) deve ser atualizada para refletir as mudanças do sistema.
* **RNF006 - Usabilidade:**
  + **RNF006.1:** O painel administrativo deve ser intuitivo, com uma curva de aprendizado baixa para usuários com conhecimento básico de sistemas.
  + **RNF006.2:** Mensagens de erro devem ser claras e orientadoras para o usuário final.
* **RNF007 - Integrabilidade:**
  + **RNF007.1:** O sistema deve ser capaz de se integrar com as APIs RESTful do DocSales, Omie e Next, respeitando seus contratos de serviço e limites de requisições.
* **RNF008 - Auditoria e Rastreabilidade:**
  + **RNF008.1:** Todas as operações críticas (ex: criação/atualização de cliente, reprocessamento de proposta) devem ser logadas com informações de usuário, data/hora e detalhes da ação.

# ****Estratégia de Integração entre Sistemas****

A estratégia de integração do SIGVoIP é baseada em uma arquitetura de Micro-serviços Orquestrados (conceitualmente, através de Application Services e Adaptadores) com comunicação via API RESTful (JSON), utilizando um modelo de integração orientada a eventos e processamento em lote.

* **Gatilho e Pull de Dados (DocSales):** O processo é iniciado por um mecanismo de polling (consulta periódica) ou webhook (futuro) que detecta novas propostas "aprovadas" no DocSales. O SIGVoIP puxa os dados completos da proposta através da API DocSales.
* **Mapeamento e Transformação:** Os dados recebidos do DocSales (em seus Integration Models) são transformados e mapeados para as entidades do **Domínio** do SIGVoIP, principalmente ClienteSIG e ServicoContratado. Este é um ponto crucial para garantir a **consistência da linguagem ubíqua** e a validação do domínio.
* **Orquestração Sequencial:** Após o mapeamento, o Application Service orquestra as chamadas sequenciais aos sistemas externos:
  + **Omie (Cadastro de Clientes e Contas a Receber):** O adaptador Omie (na camada de Infraestrutura) é invocado para cadastrar/atualizar o cliente e criar as contas a receber recorrentes.
  + **Next (Cadastro de Assinantes e Serviços):** O adaptador Next (na camada de Infraestrutura) é invocado para cadastrar/atualizar o assinante e provisionar os serviços, incluindo a lógica de seleção de servidor.
* **Armazenamento de IDs Externas: As IDs geradas pelos sistemas externos (Omie ID, Next ID) são armazenadas de volta no ClienteSIG ou ServicoContratado (como Value Objects) para futura referência e rastreabilidade.**
* **Tratamento de Erros e Reprocessamento: Erros de integração são capturados, logados detalhadamente e associados à proposta. O sistema permite o reprocessamento manual de propostas com falha, garantindo resiliência.**
* **Comunicação Bidirecional (Futuro): Embora a Fase 1 seja primariamente unidirecional (DocSales -> SIGVoIP -> Omie/Next), a arquitetura é projetada para suportar interações bidirecionais e notificações de status de volta aos sistemas de origem no futuro.**

A comunicação com as APIs externas será feita utilizando clientes HTTP robustos, com tratamento de timeouts, retentativas e logs detalhados para depuração e auditoria.

# ****Regras de Negócio****

As regras de negócio do SIGVoIP são definidas para governar o comportamento do sistema e garantir a integridade dos dados e a lógica operacional.

* **RN1:** Apenas propostas do DocSales com status "approved" (aprovado) são elegíveis para integração.
* **RN2:** Uma proposta DocSales deve ser processada apenas uma vez com sucesso para cada sistema externo (Omie, Next), a menos que explicitamente acionado para reprocessamento.
* **RN3:** Se um cliente já existe no Omie, o sistema deve atualizá-lo em vez de criar um novo. A identificação será baseada no CPF/CNPJ.
* **RN4:** Se um assinante já existe no Next, o sistema deve atualizá-lo em vez de criar um novo. A identificação será baseada em critérios definidos pela API Next.
* **RN5:** Ao provisionar serviços no Next, o sistema deve aplicar a lógica de seleção do servidor (NextBilling/NextRouter) com a menor ocupação ou conforme regras de afinidade.
* **RN6:** As contas a receber recorrentes geradas no Omie devem refletir a periodicidade, o valor e o número de parcelas especificados na proposta DocSales.
* **RN7:** Informações sensíveis (ex: senhas de acesso a APIs externas) não devem ser armazenadas em logs.
* **RN8:** Qualquer falha na integração com DocSales, Omie ou Next deve ser registrada e a proposta marcada para revisão/reprocessamento.
* **RN9:** O acesso ao painel administrativo e às funções de reprocessamento é restrito a usuários com perfil de "Administrador".
* **RN10:** A criação de contas de usuário internas para o SIGVoIP é restrita a administradores.
* **RN11:** Clientes Pessoa Jurídica (ClienteSIG com CNPJ) devem ter pelo menos um ContatoCliente associado.
* **RN12:** A validação de Value Objects (ex: DocumentoFiscal, Email, Telefone) deve ser realizada no construtor ou métodos de fábrica do próprio VO, lançando exceções específicas de domínio em caso de falha.
* **RN13:** Exceções capturadas na camada de Domínio devem ser re-lançadas como exceções de domínio customizadas, sempre encapsulando a InnerException e compondo uma mensagem clara e contextualizada.

# ****Plano de Testes****

O Plano de Testes do SIGVoIP visa garantir a qualidade, confiabilidade e conformidade do sistema com os requisitos especificados.

## Tipos de Testes:

* **Testes Unitários:**
  + **Foco:** Testar componentes individuais (classes, métodos) isoladamente.
  + **Cobertura:** Prioridade para camadas de **Domínio** e **Aplicação** (alvo > 80%).
  + **Ferramentas:** xUnit / NUnit (para .NET C#).
  + **Estratégia:** Uso extensivo de mocks e stubs para isolar dependências.
* **Testes de Integração:**
  + **Foco:** Verificar a interação entre diferentes componentes ou subsistemas.
  + **Abrangência:** Testes de integração de Domínio-Infraestrutura (ex: repositório com banco de dados de teste). Testes de integração com APIs externas (simulando ou usando ambientes de sandbox/homologação).
  + **Ferramentas:** Testes automatizados usando as mesmas ferramentas de unitários, mas com dependências reais (ex: Docker para banco de dados).
* **Testes de Sistema/End-to-End (E2E):**
  + **Foco:** Validar o fluxo completo de negócios do ponto de vista do usuário final.
  + **Abrangência:** Simular a entrada de uma proposta DocSales e verificar a correta criação no Omie e Next, além do registro no SIGVoIP.
  + **Ferramentas:** Frameworks de teste de UI (se houver Front-end) ou testes de API de alto nível.
* **Testes de Regressão:**
  + **Foco:** Garantir que novas mudanças não introduzam defeitos em funcionalidades existentes.
  + **Frequência:** Executados automaticamente no pipeline de CI/CD em cada Pull Request/Push.
* **Testes de Segurança:**
  + **Foco:** Identificar vulnerabilidades de segurança (ex: injeção de SQL, XSS, quebra de autenticação).
  + **Ferramentas:** Ferramentas de análise estática de código (SAST) e dinâmica (DAST), testes manuais de penetração (se aplicável).
  + **Diretrizes:** Baseado nas recomendações do OWASP.
* **Testes de Desempenho e Carga:**
  + **Foco:** Avaliar a capacidade do sistema sob diferentes cargas de trabalho e identificar gargalos.
  + **Ferramentas:** JMeter, K6 ou outras ferramentas de teste de carga.
  + **Métricas:** Latência de resposta, taxa de transferência, utilização de recursos.

## Ambiente de Testes:

* **Desenvolvimento:** Ambientes locais com bancos de dados de teste (Docker).
* **Integração Contínua:** Ambiente de build e teste automatizado (GitHub Actions / Azure DevOps).
* **Homologação:** Ambiente que replica a produção, conectado a sandboxes ou ambientes de teste das APIs DocSales, Omie e Next.

(Nota: Para um detalhamento mais aprofundado do plano de testes, consulte o docs/planejamento/10\_plano\_testes.md.)

# ****Roadmap de Etapas Futuras****

Após a conclusão e estabilização da Fase 1 do SIGVoIP, que foca na automação essencial do fluxo de propostas DocSales para Omie e Next, o projeto tem uma visão clara de expansão e aprimoramento. As próximas etapas prioritárias incluem:

* **Fase 2 - Automação Bidirecional e Dashboard Avançado:**
  + Implementação de Webhooks no DocSales para notificações de propostas aprovadas em tempo real, reduzindo a latência da integração.
  + Sincronização de status de serviços e faturamento de volta para o DocSales/Omie.
  + Desenvolvimento de um dashboard administrativo mais robusto no SIGVoIP, com relatórios detalhados, métricas de integração e alertas proativos.
  + Automação de processos de alteração e cancelamento de serviços em todos os sistemas envolvidos.
* **Fase 3 - Expansão de Integrações e Inteligência:**
  + Avaliação e integração com outras plataformas de CRM/ERP/VoIP, conforme necessidade de negócio.
  + Implementação de lógica de decisão baseada em IA/ML para otimização da alocação de recursos no Next (ex: balanceamento de carga inteligente).
  + Módulos de auditoria e conformidade mais aprofundados.
* **Melhorias Contínuas:**
  + Otimização de performance e escalabilidade.
  + Atualização de tecnologias e frameworks para versões mais recentes.
  + Expansão da cobertura de testes automatizados e melhoria dos pipelines de CI/CD.

O detalhamento e a priorização dessas fases serão definidos em novos planos de desenvolvimento, com base no feedback dos usuários e nas necessidades estratégicas da empresa.

# ****Glossário de Termos****

* **ADR (Architectural Decision Record):** Documento curto que captura uma decisão arquitetural importante e seu contexto.
* **Aggregate Root (Raiz do Agregado):** Entidade principal de um cluster de objetos de domínio, que garante a consistência transacional do agregado.
* **API (Application Programming Interface):** Conjunto de definições e protocolos que permite que diferentes softwares se comuniquem.
* **Application Service (Serviço de Aplicação):** Orquestra os casos de uso, mediando entre a interface e a lógica de domínio.
* **Clean Architecture:** Estilo arquitetural que prioriza a lógica de negócio (domínio) no centro, independente de frameworks e tecnologias externas.
* **CRM (Customer Relationship Management):** Sistema de gestão de relacionamento com o cliente (ex: DocSales).
* **DER (Diagrama Entidade-Relacionamento):** Representação visual da estrutura de um banco de dados relacional.
* **DDD (Domain-Driven Design):** Abordagem de desenvolvimento de software que foca na modelagem de um domínio de negócio complexo.
* **DIP (Dependency Inversion Principle):** Princípio SOLID que prega que módulos de alto nível não devem depender de módulos de baixo nível; ambos devem depender de abstrações.
* **DocSales:** Sistema CRM externo utilizado para gestão de propostas.
* **Domínio:** A camada mais interna da arquitetura, contendo as regras de negócio e a lógica essencial.
* **DTO (Data Transfer Object):** Objeto simples usado para transferir dados entre camadas ou sistemas.
* **ERP (Enterprise Resource Planning):** Sistema de gestão integrada de processos empresariais (ex: Omie).
* **Fase 1:** A etapa inicial do projeto, focada nas funcionalidades de automação de propostas DocSales para Omie e Next.
* **Hexagonal Architecture (Ports and Adapters):** Estilo arquitetural que isola a lógica de domínio de detalhes técnicos, usando "portas" (interfaces) e "adaptadores" (implementações).
* **IntegMod (Integration Model):** Estrutura de dados específica utilizada para interagir com APIs de sistemas externos, espelhando seus contratos.
* **JWT (JSON Web Token):** Padrão aberto para criação de tokens de acesso, usado para autenticação e autorização.
* **Kanban:** Framework ágil para gestão de fluxo de trabalho.
* **Linguagem Ubíqua:** Vocabulário comum entre desenvolvedores e especialistas de domínio.
* **MySQL:** Sistema de gerenciamento de banco de dados relacional.
* **NextBilling / NextRouter:** Plataformas de telefonia VoIP externas para provisionamento de serviços.
* **Omie:** Sistema ERP externo.
* **OWASP (Open Web Application Security Project):** Comunidade online focada em segurança de software.
* **Regra de Dependência:** Princípio da Clean Architecture onde as dependências de código apontam sempre para as camadas mais internas.
* **Repositório:** Padrão de design que abstrai a lógica de persistência de dados.
* **SIGVoIP:** Sistema de Integração e Gestão para Telefonia VoIP (nome do projeto).
* **SOLID:** Conjunto de cinco princípios de design de software para tornar os designs mais compreensíveis, flexíveis e manuteníveis.
* **Value Object (Objeto de Valor):** Objeto imutável que representa um conceito do domínio definido por seus atributos (ex: Endereço, Dinheiro).
* **VoIP (Voice over Internet Protocol):** Tecnologia que permite fazer chamadas de voz e vídeo pela internet.

# ****Histórico de Versões da Documentação****

| Versão | Data | Autor | Descrição da Mudança |
| --- | --- | --- | --- |
| 1.0.0 | 02/07/2025 | Leandro Rogerio | Criação inicial do Documento Formal do Sistema SIGVoIP, com inclusão do Resumo Executivo e Roadmap de Etapas Futuras. |