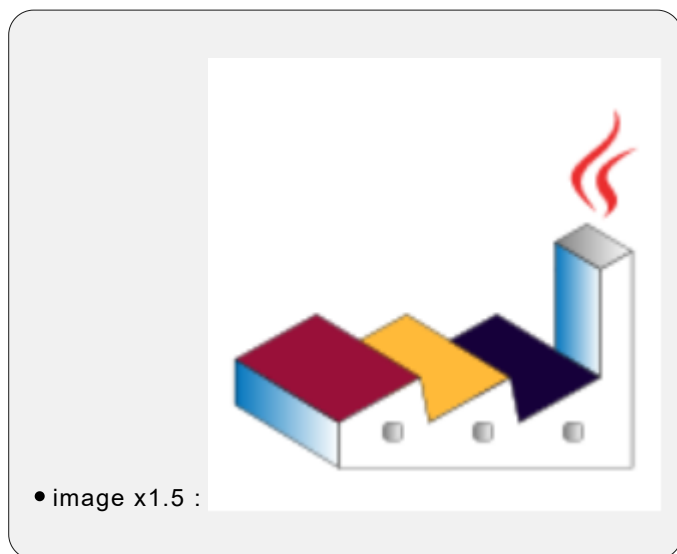


# Overview

---



## Descrição dos Requisitos Funcionais e Não Funcionais e Considerações Arquiteturais

---

### Requisitos Funcionais:

1. O sistema deve permitir o cadastro de clientes com informações como nome, endereço e tipo de serviço.
2. O sistema deve permitir o registro das leituras dos medidores de gás, água e luz.
3. O sistema deve calcular o consumo dos clientes com base nas leituras registradas.
4. O sistema deve gerar relatórios de consumo para os clientes, contendo informações detalhadas sobre seu consumo de serviços.
5. O sistema deve fornecer aos clientes uma segunda via de boletos para pagamento dos serviços.
6. O sistema deve permitir que os administradores de condomínio acessem relatórios de consumo dos clientes vinculados ao condomínio.
7. O sistema deve enviar notificações por e-mail aos clientes sobre atualizações de consumo e vencimento de boletos.

### Requisitos Não Funcionais:\*\*

1. O sistema deve garantir a segurança das informações dos clientes, adotando medidas adequadas de proteção de dados.
2. O sistema deve ser de fácil utilização e possuir uma interface intuitiva para os usuários.
3. O sistema deve ser escalável, permitindo o crescimento do número de clientes e registros de leitura.
4. O sistema deve ser robusto e tolerante a falhas, garantindo a disponibilidade dos serviços.
5. O sistema deve ter um desempenho eficiente, processando as operações de cadastro, leitura e cálculo de consumo de forma rápida e eficaz.

6. O sistema deve ser compatível com diferentes dispositivos e navegadores, garantindo a acessibilidade para os usuários.

## Considerações Arquiteturais

### Escalabilidade

Uma consideração importante na arquitetura do sistema de leitura de medidores de gás, água e luz é a escalabilidade. Para atingir a escalabilidade, o sistema será projetado com uma arquitetura distribuída e escalável, capaz de lidar com um grande volume de leituras e usuários. Será adotado o dimensionamento horizontal, permitindo adicionar novos servidores de acordo com a demanda.

### Segurança

A segurança é uma preocupação essencial no sistema, pois envolve dados sensíveis dos clientes. Para garantir a segurança, serão implementadas medidas como criptografia de dados, controle de acesso baseado em papéis e auditoria de atividades. Serão adotadas práticas de segurança recomendadas para proteger a integridade, confidencialidade e autenticidade dos dados.

### Resiliência

A resiliência do sistema é fundamental para garantir a disponibilidade contínua dos serviços. Serão implementadas estratégias de recuperação de falhas, como o uso de servidores redundantes, balanceamento de carga e monitoramento constante da saúde do sistema. Dessa forma, o sistema poderá se recuperar rapidamente de falhas e minimizar o impacto nos usuários.

### Integração

O sistema de leitura de medidores precisa se integrar com outros sistemas, como o ERP externo e o sistema de envio de e-mails. Serão fornecidas interfaces adequadas para facilitar a integração eficiente e confiável com esses sistemas. Serão utilizados padrões e protocolos apropriados para garantir a compatibilidade e a transferência de dados sem problemas.

### Manutenibilidade

Considerações de manutenibilidade são importantes para garantir a facilidade de manutenção e evolução do sistema ao longo do tempo. A arquitetura será projetada de forma modular e bem documentada, facilitando a compreensão e modificação dos componentes individuais. A automação de testes e implantação contínua será adotada para agilizar o processo de manutenção. Serão seguidas boas práticas de desenvolvimento e documentação para garantir a manutenibilidade do sistema.

### Inversão de Controle

Será adotado o princípio de inversão de controle (IoC) para desacoplamento e maior flexibilidade. Através da inversão de controle, as dependências entre os componentes serão gerenciadas externamente, permitindo que as implementações possam ser substituídas facilmente. Isso proporcionará maior modularidade, facilidade de teste e reutilização de código.

### TDD e DDD

Será utilizado o desenvolvimento orientado a testes (TDD) como prática de desenvolvimento. Isso envolve escrever os testes antes de implementar o código, garantindo que o sistema seja testável desde o início. Além disso, será adotado o design orientado a domínio (DDD) para garantir que o sistema reflita corretamente as regras de negócio e mantenha uma linguagem ubíqua. O DDD ajudará a organizar o código em torno dos conceitos do domínio, facilitando a compreensão e manutenção.

## Princípios SOLID

Os princípios SOLID serão seguidos durante o desenvolvimento do sistema. Cada componente será projetado com base nos princípios de Responsabilidade Única (SRP), Aberto/Fechado (OCP), Substituição de Liskov (LSP), Segregação de Interfaces (ISP) e Inversão de Dependência (DIP). Isso garantirá que o sistema seja modular, flexível, fácil de entender e manter.