



CisternaGuardianPB

Monitoramento e gerenciamento de cisternas de água em comunidades do estado da Paraíba

Documento de escopo do projeto.

Equipe de Engenharia do Projeto Cooperar

- Engenheiro Civil: Kleber de Sousa Batista
- Engenheiro Civil: João Guilherme Quirino Nelo
- Engenheiro de Computação: Iury Anderson Fernandes Coelho

TABELA DE REVISÕES		
Data da Revisão	Revisor	Alteração Realizada
15/02/2023	Iury Coelho	Versão inicial do documento
10/03/2023	Iury Coelho	Adicionados requisitos funcionais e não-funcionais
15/03/2023	Iury Coelho	Descrição da API detalhada
17/03/2023	Iury Coelho	Adicionada arquitetura geral do sistema

Sumário

Introdução	3
Metas e objetivos do projeto	4
Requisitos do Sistema CisternaGuardianPB	5
Requisitos não funcionais	7
Escopo Geral do Projeto	8
Arquitetura do Sistema	9
Camada de Integração	10
Definição do objetivo e escopo da API	11
Escolha da linguagem e framework	11
Implementação das rotas e operações	13
Rotas da API REST do Sistema	13
Rotas de Cisternas	14

Introdução

O projeto CisternaGuardian PB tem como objetivo desenvolver um sistema completo de monitoramento e gerenciamento de cisternas de água em comunidades do estado da Paraíba. Para atender a este objetivo, o sistema deve ser desenvolvido utilizando tecnologias como Angular, Spring Boot, MongoDB e APIs REST.

O documento de escopo do projeto apresenta as principais características e funcionalidades do sistema, bem como a arquitetura do sistema, incluindo as camadas de apresentação, aplicação, persistência, integração e segurança. Além disso, o documento define as rotas e operações da API REST do sistema, que permitirá que terceiros acessem e manipulem os dados do sistema de maneira segura e controlada.

Este documento é destinado a todos os envolvidos no projeto, incluindo a equipe de desenvolvimento. Ele descreve o escopo geral do projeto, as metas e objetivos do projeto e os requisitos técnicos e funcionais do sistema. Ele é um documento vivo e será atualizado conforme novas informações e requisitos forem identificados.

Metas e objetivos do projeto

As metas e objetivos do projeto Guardian Cisterna são:

1. Desenvolver um sistema completo de monitoramento e gerenciamento de cisternas de água em comunidades do estado da Paraíba.
2. Fornecer aos usuários finais do sistema informações precisas e em tempo real sobre o nível de água e qualidade da água armazenada nas cisternas.
3. Facilitar a manutenção preventiva e corretiva das cisternas, reduzindo os custos de manutenção e aumentando a vida útil das mesmas.
4. Permitir que terceiros acessem e manipulem os dados do sistema por meio de uma API REST segura e controlada.
5. Utilizar tecnologias modernas e escaláveis como Angular, Spring Boot, MongoDB e APIs REST para o desenvolvimento do sistema.
6. Garantir a segurança dos dados e a privacidade dos usuários finais do sistema, atendendo às normas e regulamentações aplicáveis.

Essas metas e objetivos foram definidos com base nas necessidades e expectativas dos usuários finais do sistema. O objetivo geral é fornecer um sistema de monitoramento e gerenciamento de cisternas eficiente, confiável e seguro, que ajude a melhorar a qualidade de vida das comunidades atendidas.

Requisitos do Sistema CisternaGuardianPB

Lista detalhada dos requisitos técnicos e funcionais do sistema CisternaGuardianPB:

Requisitos técnicos

1. Hardware: o sistema pode exigir um servidor dedicado ou uma nuvem para armazenar os dados e executar as funcionalidades do sistema. Além disso, pode ser necessário um conjunto de sensores para monitorar o local da obra e coletar dados em tempo real.
2. Software: o sistema pode exigir um sistema operacional, como o Linux ou o Windows, e um banco de dados para armazenar os dados. Além disso, pode ser necessário um conjunto de bibliotecas e frameworks de software para implementar as funcionalidades do sistema.
3. Integração: o sistema pode precisar integrar-se com outros sistemas relevantes, como sistemas de gestão de projetos, sistemas de gerenciamento de recursos humanos e outros sistemas de monitoramento e fiscalização.
4. Segurança: o sistema deve garantir a proteção dos dados armazenados e assegurar a privacidade dos usuários. Isso pode incluir medidas de segurança como criptografia de dados, autenticação de usuário e controle de acesso.

Requisitos funcionais

1. Monitoramento do progresso da obra: o sistema deve permitir acompanhar o progresso da obra em tempo real, incluindo o uso de materiais e mão de obra, o cumprimento de prazos e o cumprimento de padrões de qualidade.
2. Verificação de cumprimento de requisitos: o sistema deve permitir verificar o cumprimento dos requisitos exigidos pelo projeto, incluindo o cumprimento de padrões de qualidade, o uso de materiais adequados e o cumprimento de prazos.
3. Gerenciamento de equipe: o sistema deve permitir gerenciar as responsabilidades da equipe envolvida na construção das cisternas, garantindo que cada membro da equipe saiba o que é esperado dele e possa cumprir suas tarefas de maneira eficiente.

4. Relatórios detalhados: o sistema deve permitir gerar relatórios detalhados sobre o progresso da obra, o uso de recursos e outros aspectos importantes do projeto, o que pode ser útil para a tomada de decisões e para a avaliação do projeto.
5. Alertas de segurança: o sistema deve monitorar o local da obra e enviar alertas em caso de emergência ou de condições de trabalho perigosas, garantindo a segurança dos trabalhadores e das comunidades beneficiadas pelas cisternas.
6. Atendimento às necessidades da comunidade: o sistema deve permitir acompanhar as necessidades da comunidade beneficiada pelas cisternas e garantir que elas sejam atendidas de maneira adequada. Isso pode incluir o acompanhamento da qualidade da água armazenada, a identificação de problemas com a cisterna e a resolução de eventuais problemas.
7. Integração com outros sistemas: o sistema deve ser capaz de se integrar com outros sistemas relevantes, como sistemas de gestão de projetos, sistemas de gerenciamento de recursos humanos e outros sistemas de monitoramento e fiscalização.
8. Acesso a informações em tempo real: o sistema deve permitir que os usuários acessem facilmente as informações relevantes sobre as cisternas e o progresso da obra, de maneira a facilitar a tomada de decisões e a resolução de problemas.
9. Análise de gráficos: o sistema deve permitir gerar gráficos e outras visualizações de dados que possam ajudar a entender o progresso da obra, o uso de recursos e outros aspectos importantes do projeto.
10. Aceleração do processo de entrega de recursos: o sistema deve permitir avaliar de maneira mais rápida o cumprimento dos requisitos exigidos pelo projeto, incluindo o cumprimento de padrões de qualidade, acelerando o processo de entrega de recursos às empresas licitadas.
11. Gerenciamento de documentação: o sistema deve permitir armazenar e gerenciar documentos relevantes para o projeto, como contratos, relatórios técnicos e outros documentos.

Requisitos não funcionais

1. Disponibilidade: o sistema deve estar disponível para uso a maior parte do tempo, com tempo de inatividade mínimo para manutenção ou atualizações.
2. Escalabilidade: o sistema deve ser capaz de lidar com um aumento na demanda de usuários ou dados sem perda de desempenho.
3. Segurança: o sistema deve proteger os dados dos usuários e garantir a privacidade das informações. Isso pode incluir medidas de segurança como autenticação de usuário, criptografia de dados e proteção contra ataques cibernéticos.
4. Desempenho: o sistema deve responder rapidamente às solicitações dos usuários e carregar páginas de maneira rápida e eficiente.
5. Compatibilidade: o sistema deve ser compatível com diferentes plataformas e navegadores de internet.
6. Facilidade de uso: o sistema deve ser fácil de usar e compreender para todos os usuários, independentemente de sua habilidade técnica.
7. Testabilidade: o sistema deve ser fácil de testar e validar para garantir sua qualidade e confiabilidade.
8. Manutenção: o sistema deve ser fácil de manter e atualizar para garantir sua disponibilidade e desempenho a longo prazo.

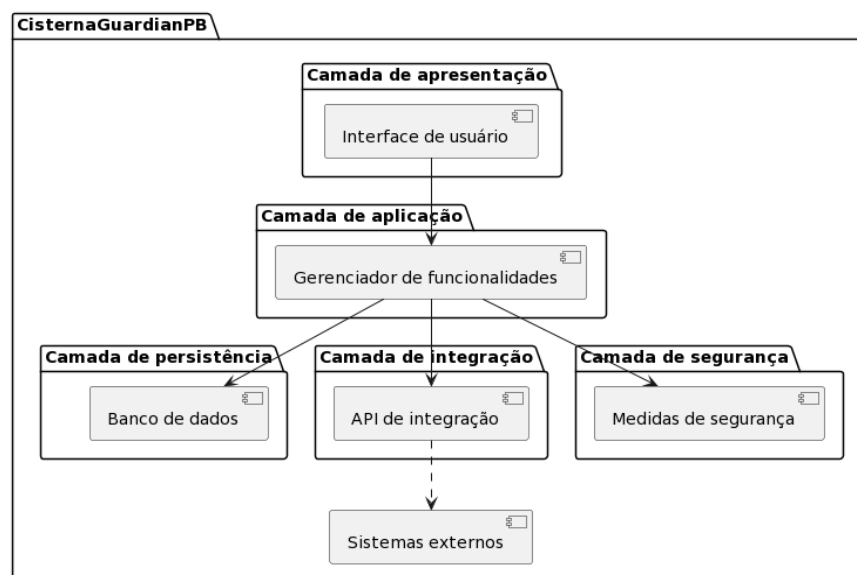
Escopo Geral do Projeto

O escopo do projeto Guardian Cisterna é desenvolver um sistema completo de monitoramento e gerenciamento de cisternas de água em comunidades do estado da Paraíba. O sistema deve atender a uma série de requisitos técnicos e funcionais, incluindo hardware, software, integração, segurança e disponibilidade. Além disso, deve ser desenvolvido uma arquitetura do sistema que inclui as camadas de apresentação, aplicação, persistência, integração e segurança. O sistema deve permitir o monitoramento do progresso da obra, a verificação do cumprimento dos requisitos, o gerenciamento da equipe, a geração de relatórios detalhados, a emissão de alertas de segurança, o atendimento às necessidades da comunidade, a integração com outros sistemas, a análise de gráficos, a aceleração do processo de entrega de recursos e o gerenciamento de documentação. O sistema também deve atender a requisitos não funcionais, como escalabilidade, segurança, desempenho, compatibilidade, facilidade de uso, testabilidade e manutenção. O sistema deve ser implementado utilizando tecnologias como Angular, Spring Boot, MongoDB e APIs REST.

Arquitetura do Sistema

A arquitetura do sistema CisternaGuardianPB inclui as seguintes camadas:

1. Camada de apresentação: esta camada é responsável por fornecer a interface de usuário do sistema, permitindo que os usuários acessem as funcionalidades do sistema através de um navegador web. A camada de apresentação é implementada utilizando tecnologias como HTML, CSS e JavaScript juntamente com o framework Angular.
2. Camada de aplicação: esta camada é responsável por implementar as funcionalidades do sistema e por gerenciar a lógica de negócio. Ela é implementada utilizando frameworks de aplicação Java Spring Boot.
3. Camada de persistência: esta camada é responsável por armazenar e recuperar os dados do sistema, utilizando um banco de dados. Deve ser implementada utilizando tecnologias NoSQL no MongoDB.
4. Camada de integração: esta camada é responsável por integrar o sistema com outros sistemas relevantes, como sistemas de gestão de projetos, sistemas de gerenciamento de recursos humanos e outros sistemas de monitoramento e fiscalização. Deve ser implementada utilizando tecnologias de integração APIs REST.
5. Camada de segurança: esta camada é responsável por garantir a proteção dos dados armazenados e a privacidade dos usuários. Deve ser implementada utilizando medidas de segurança como criptografia de dados, autenticação de usuário e controle de acesso.



Camada de Integração

A integração do sistema será feita através de um API Rest escrito em Java utilizando o framework Spring Boot. A estrutura da API Rest do sistema CisternaGuardianPB inclui os seguintes módulos ou funcionalidades:

1. Autenticação e autorização: esta funcionalidade é responsável por garantir que apenas usuários autorizados tenham acesso às funcionalidades da API. Ela pode ser implementada utilizando técnicas de autenticação, como o uso de tokens de acesso ou credenciais de usuário.
2. Gerenciamento de usuários: esta funcionalidade é responsável por permitir que os usuários sejam criados, atualizados, excluídos e consultados na base de dados. Ela pode ser implementada através de rotas HTTP como POST, GET, PUT e DELETE.
3. Gerenciamento de obras: esta funcionalidade é responsável por permitir que as obras de cisternas sejam criadas, atualizadas, excluídas e consultadas na base de dados. Ela também deve permitir acompanhar o progresso das obras e identificar problemas ou atrasos.
4. Gerenciamento de beneficiários: esta funcionalidade é responsável por permitir que os beneficiários das cisternas sejam cadastrados, atualizados, excluídos e consultados na base de dados. Ela também deve permitir acompanhar as necessidades da comunidade beneficiada pelas cisternas e garantir que elas sejam atendidas de maneira adequada.
6. Gerenciamento de documentos: esta funcionalidade é responsável por permitir que os documentos relevantes para o projeto sejam armazenados e gerenciados, incluindo contratos, relatórios técnicos e outros documentos.
7. Notificações: esta funcionalidade é responsável por enviar notificações por e-mail ou outros meios para os usuários relevantes sobre alterações no progresso da obra, alertas de segurança e outras informações importantes.

Definição do objetivo e escopo da API

O objetivo da API do sistema CisternaGuardianPB é fornecer uma interface de programação de aplicativos (API) para permitir que terceiros acessem e manipulem os dados do sistema de maneira segura e controlada.

O escopo da API inclui a exposição de dados relacionados a obras, cisternas, beneficiários, fiscalizações e outras entidades relevantes do sistema, bem como a disponibilização de funcionalidades para a realização de operações de cadastro, consulta, atualização e exclusão desses dados.

A API também deve prover autenticação e autorização de acesso para garantir que somente os usuários autorizados tenham acesso aos dados e funcionalidades permitidas. A API também deve permitir o upload e download de arquivos, como fotos e relatórios técnicos, e fornecer mecanismos de autenticação e autorização para garantir a segurança dos dados.

Escolha da linguagem e framework

Spring Boot é um framework para o desenvolvimento de aplicações Java com configuração automatizada e sem a necessidade de um arquivo de configuração complexo. Ele oferece uma série de recursos úteis para o desenvolvimento de APIs REST, como gerenciamento de dependências, configuração de rotas e endpoints, integração com banco de dados e validação de entrada de dados.

Algumas vantagens de utilizar Java com Spring Boot para desenvolver a API do CisternaGuardianPB são:

1. Vasta gama de bibliotecas: Java possui uma ampla gama de bibliotecas disponíveis, o que facilita a integração com outras ferramentas e sistemas.
2. Configuração automatizada: Spring Boot simplifica a configuração da aplicação, permitindo que o desenvolvedor se concentre no desenvolvimento da lógica da aplicação em vez de se preocupar com a configuração complexa.
3. Recursos úteis para o desenvolvimento de APIs: Spring Boot oferece vários recursos úteis para o desenvolvimento de APIs REST, como gerenciamento de dependências, configuração de rotas e endpoints, integração com banco de dados e validação de entrada de dados.

Definição da arquitetura da API

A arquitetura da API do CisternaGuardianPB pode ser dividida em três camadas principais:

1. Camada de Entidade: nesta camada, são definidas as classes de entidade que representam os objetos do sistema. Por exemplo, podemos ter uma classe Obra que

representa uma obra de cisterna, uma classe Beneficiário que representa um beneficiário de cisterna, etc. As classes de entidade são anotadas com `@Entity` e são persistidas no banco de dados através de um repositório de acesso a dados, como o MongoDB Repository.

2. Camada de Serviço: nesta camada, são definidos os serviços que realizam as operações lógicas do sistema. Por exemplo, podemos ter um serviço `ObraService` que realiza operações de CRUD (criação, leitura, atualização e exclusão) de obras de cisterna, um serviço `BeneficiarioService` que realiza operações de CRUD de beneficiários, etc. Os serviços são anotados com `@Service` e são responsáveis por realizar as operações lógicas do sistema, como validações, regras de negócio, etc.
3. Camada de Controlador: nesta camada, são definidos os controladores que expõem a API REST do sistema. Por exemplo, podemos ter um controlador `ObraController` que expõe rotas para criar, ler, atualizar e excluir obras de cisterna, um controlador `BeneficiarioController` que expõe rotas para criar, ler, atualizar e excluir beneficiários, etc. Os controladores são anotados com `@Controller` e são responsáveis por receber as requisições HTTP, chamar os serviços adequados para realizar as operações lógicas e retornar a resposta ao cliente.

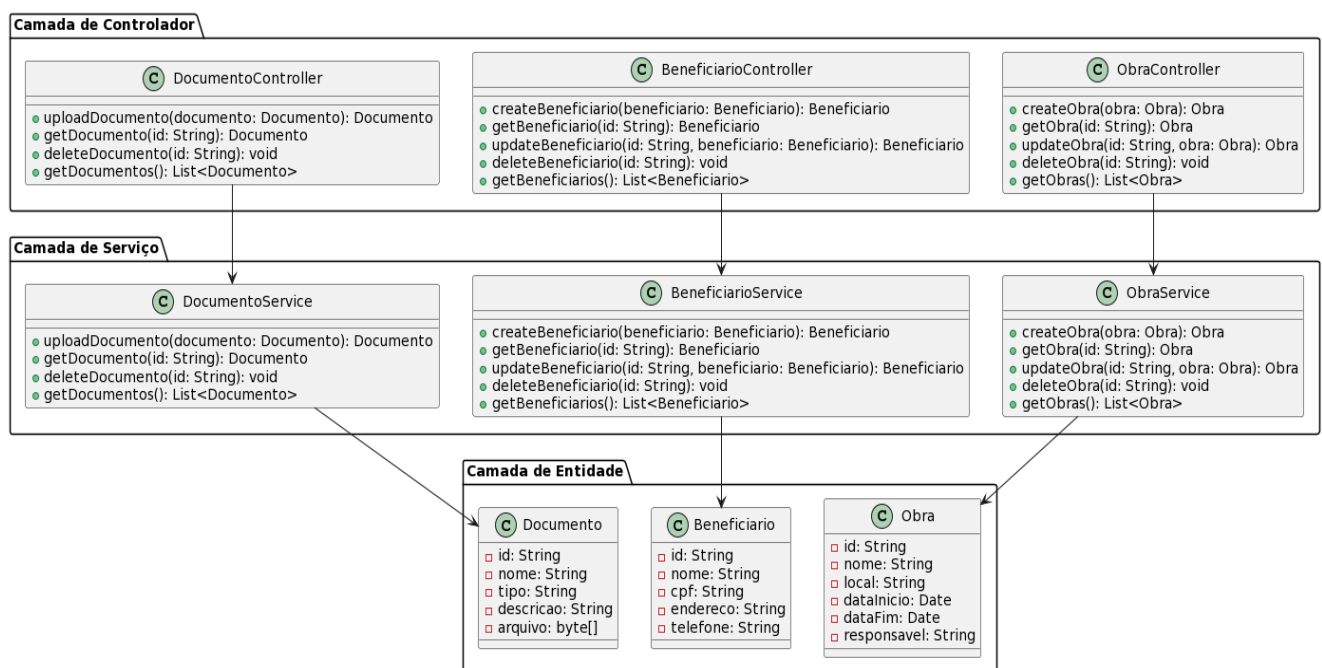


Figura 1 - Fluxograma da arquitetura da API do sistema CisternaGuardianPB.

Implementação das rotas e operações

A implementação das rotas e operações do sistema CisternaGuardianPB será realizada através da camada de controllers. Cada controller será responsável por gerenciar as

requisições HTTP para uma determinada entidade do sistema, como obras, cisternas, beneficiários, etc.

As rotas serão implementadas seguindo o padrão REST, com os métodos HTTP GET, POST, PUT e DELETE para realizar as operações de listagem, cadastro, atualização e exclusão, respectivamente. Além disso, serão incluídos métodos adicionais, como o GET para obter detalhes de uma entidade específica, por exemplo.

Cada controller conterá os métodos necessários para atender às requisições HTTP para a respectiva entidade, fazendo uso dos serviços da camada de service para realizar as operações de negócio e da camada de repository para acessar o banco de dados.

As respostas das operações serão devolvidas ao cliente no formato JSON, contendo os dados solicitados ou mensagens de erro em caso de falha. Serão tratados os possíveis erros e exceções para garantir a consistência e integridade dos dados.

Rotas da API REST do Sistema

Rotas de Obras

Método	Rota	Descrição
GET	/api/obras	lista todas as obras cadastradas.
POST	/api/obras	cadastrar uma nova obra no sistema.
GET	/api/obras/{id}	obtem os detalhes de uma obra específica, pelo seu ID.
PUT	/api/obras/{id}	atualiza os dados de uma obra específica, pelo seu ID.
DELETE	/api/obras/{id}	exclui uma obra específica, pelo seu ID.

Essas operações permitem que o sistema CisternaGuardianPB tenha as seguintes funcionalidades:

1. Listar todas as cisternas: permitir que o usuário visualize a lista de todas as cisternas cadastradas no sistema, incluindo informações como o ID, localização, beneficiário e status da cisterna.
2. Cadastrar uma nova cisterna: permitir que o usuário insira os dados de uma nova cisterna no sistema, como o ID do beneficiário, localização, especificações e status da cisterna.
3. Obter detalhes de uma cisterna: permitir que o usuário visualize os detalhes de uma cisterna específica, incluindo informações como o ID, localização, beneficiário, especificações e status da cisterna.
4. Atualizar dados de uma cisterna: permitir que o usuário altere os dados de uma cisterna específica, como o ID do beneficiário, localização, especificações e status da cisterna.
5. Excluir uma cisterna: permitir que o usuário exclua uma cisterna específica do sistema, incluindo todas as informações relacionadas à cisterna, como o ID, localização, beneficiário, especificações e status da cisterna.

Rotas de Cisternas

Para implementar as operações da rota cisterna no sistema CisternaGuardianPB, pode-se utilizar as seguintes rotas:

Método	Rota	Descrição
--------	------	-----------

GET	/api/cisternas	lista todas as cisternas cadastradas no sistema
POST	/api/cisternas	cadastra uma nova cisterna no sistema.
GET	/api/cisternas/{id}	obtem os detalhes de uma cisterna especifica, pelo seu ID.
PUT	/api/cisternas/{id}	atualiza os dados de uma cisterna especifica, pelo seu ID.
DELETE	/api/cisternas/{id}	exclui uma cisterna especifica, pelo seu ID.
GET	/api/cisternas/{id}/checklist_fiscalizacao	obtem o checklist de fiscalizacao de uma cisterna especifica.
PUT	/api/cisternas/{id}/checklist_fiscalizacao	atualiza o checklist de fiscalizacao de uma cisterna especifica.
POST	/api/cisternas/{id}/arquivos	adiciona um arquivo relacionado à cisterna especifica.
DELETE	/api/cisternas/{id}/arquivos/{arquivo id}	adiciona um arquivo relacionado à cisterna especifica.

Essas operações permitem que o sistema CisternaGuardianPB tenha as seguintes funcionalidades:

1. Listar todas as cisternas: permitir que o usuário visualize a lista de todas as cisternas cadastradas no sistema, incluindo informações como o ID, localização, beneficiário e status da cisterna.
2. Cadastrar uma nova cisterna: permitir que o usuário insira os dados de uma nova cisterna no sistema, como o ID do beneficiário, localização, especificações e status da cisterna.
3. Obter detalhes de uma cisterna: permitir que o usuário visualize os detalhes de uma cisterna especifica, incluindo informações como o ID, localização, beneficiário, especificações e status da cisterna.
4. Atualizar dados de uma cisterna: permitir que o usuário altere os dados de uma cisterna especifica, como o ID do beneficiário, localização, especificações e status da cisterna.
5. Excluir uma cisterna: permitir que o usuário exclua uma cisterna especifica do sistema, incluindo todas as informações relacionadas à cisterna, como o ID, localização, beneficiário, especificações e status da cisterna.

Para implementar a rota /cisternas no sistema CisternaGuardianPB, é necessário seguir os seguintes passos:

1. Criar a entidade Cisterna, com os atributos necessários para armazenar as informações da cisterna, como o tipo de cisterna, as especificações, a capacidade, entre outros.
2. Criar o repositório de Cisterna, utilizando a interface MongoRepository, para realizar as operações de CRUD na coleção de cisternas no banco de dados.
3. Criar o service de Cisterna, responsável por chamar os métodos do repositório e realizar as lógicas de negócio, como validações e cálculos.
4. Criar o controller de Cisterna, que será o ponto de entrada da API para a rota /cisternas. Neste controller, devem ser implementados os métodos HTTP para as operações de listagem, cadastro, atualização e exclusão de cisternas.