
Documentação de Projeto

para o sistema

Santo Restauro

Versão 1.0

Projeto de sistema elaborado pelo aluno João Vitor Santana Lopes
como parte da disciplina **Projeto de Software**.

16/11/2025

Tabela de Conteúdo

1. Introdução

2. Modelos de Usuário e Requisitos

2.1 Descrição de Atores

2.2 Modelo de Casos de Uso e Histórias de Usuários

2.3 Diagrama de Sequência do Sistema e Contrato de Operações

3. Modelos de Projeto

3.1 Arquitetura

3.2 Diagrama de Componentes e Implantação.

3.3 Diagrama de Classes

3.4 Diagramas de Sequência

3.5 Diagramas de Comunicação

3.6 Diagramas de Estados

4. Modelos de Dados

Histórico de Revisões

Nome	Data	Razões para Mudança	Versão

1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema santo restauro. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de especificação que descreve a visão de domínio do sistema.

2. Modelos de Usuário e Requisitos

2.1 Descrição de Atores

Nesta subseção é apresentado descrição de cada um dos atores que interagem com o sistema Santo Restauro.

Ator: Usuário Visitante

É o ator base do sistema, representando qualquer pessoa que acessa a plataforma sem estar autenticada.

As principais interações deste ator com o sistema são:

Visualizar Acervos Públicos: Pode navegar e consultar os acervos e itens que estão marcados como "públicos" pelo sistema.

Fazer Login: Pode se autenticar na plataforma para acessar funcionalidades restritas ao seu nível de permissão.

Ator: Detentor do Acervo

É um Usuário que se autenticou e possui um tipo de perfil específico que lhe concede permissões de gerenciamento. Este ator é, tipicamente, o proprietário ou responsável por uma coleção específica.

Suas interações são restritas apenas ao(s) seu(s) próprio(s) acervo(s):

Gerir seu Acervo: Realizar o gerenciamento completo (Criar, Ler, Atualizar, Deletar - CRUD) das informações do seu acervo.

Gerir Itens do Acervo: Gerenciar os itens pertencentes ao seu acervo, utilizando os templates de campos definidos.

Exportar Acervo: Pode exportar os dados do seu acervo para os formatos PDF ou XLSX, como uma extensão do caso de uso "gerir seu acervo".

Ator: Administrador (Adm)

É um Usuário que se autenticou e possui o tipo de perfil "administrador", concedendo-lhe acesso irrestrito a todas as funcionalidades e dados do sistema.

Suas interações são globais e incluem:

Gerir Acervos: Realizar o gerenciamento (CRUD) de todos os acervos cadastrados no sistema, incluindo os de outros detentores.

Gerir Itens de um Acervo: Gerenciar os itens de qualquer acervo.

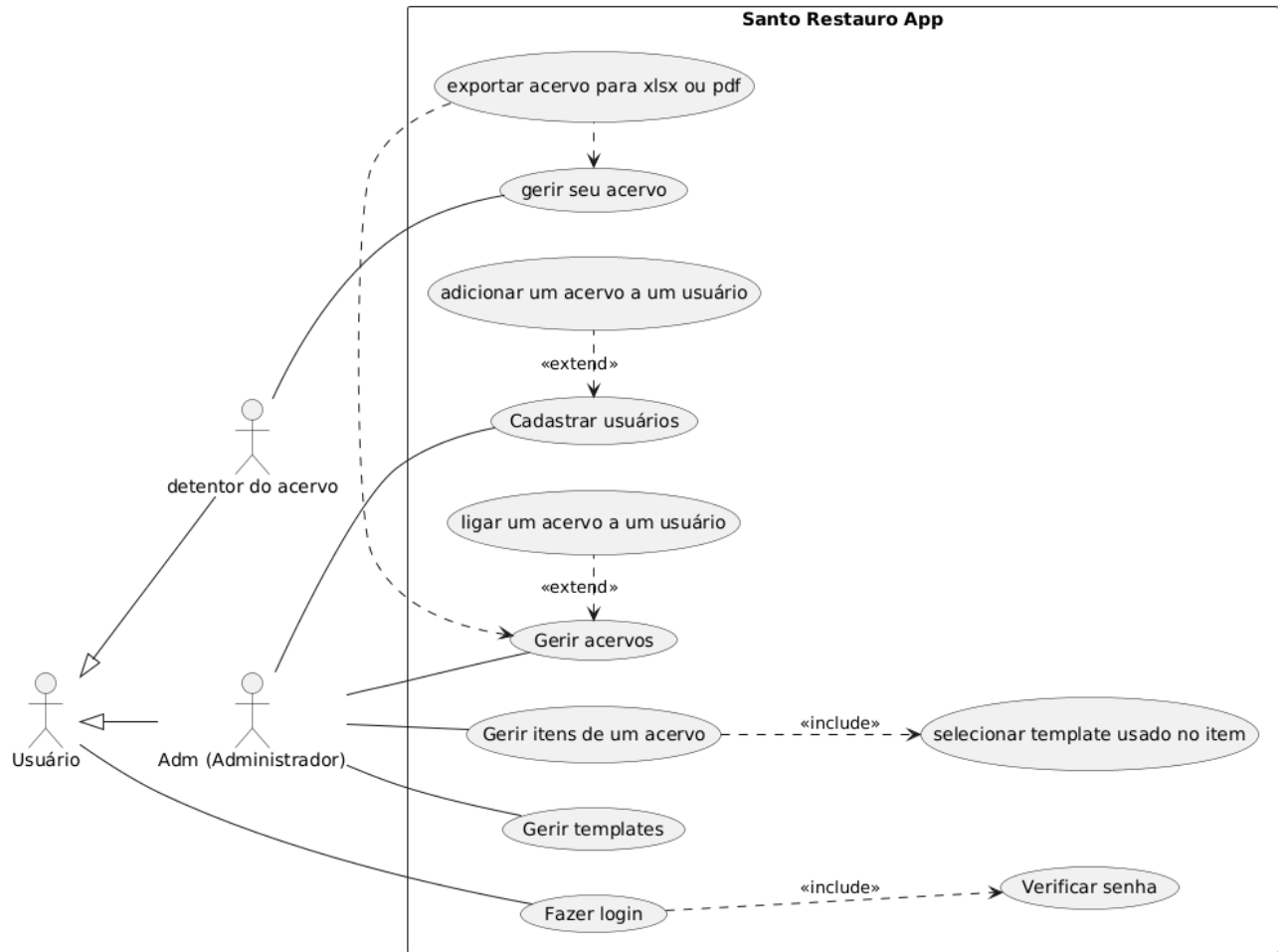
Gerir Templates: Criar, editar e excluir os templates que definem os campos de dados dos itens.

Cadastrar Usuários: Criar novas contas de usuário, definindo seus tipos (Administrador ou Detentor do Acervo).

Ligar Acervo a um Usuário: Associar um acervo existente a um usuário (Detentor) específico.

2.2 Modelo de Casos de Uso

Nesta subseção é apresentado o diagrama de casos de uso do sistema.

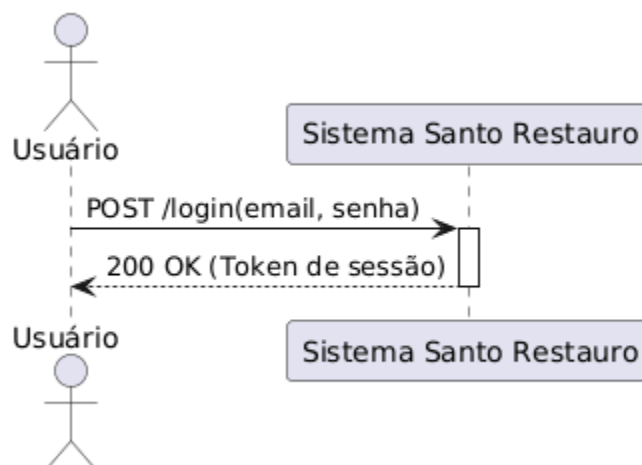


2.3 Diagrama de Sequência do Sistema

Nesta subseção é apresentado o diagrama de sequência do sistema de 3 Casos de Uso principais. Estes diagramas ilustram a interação de alto nível entre o Ator e o Sistema, tratando o sistema como uma entidade única ("caixa preta"), conforme solicitado.

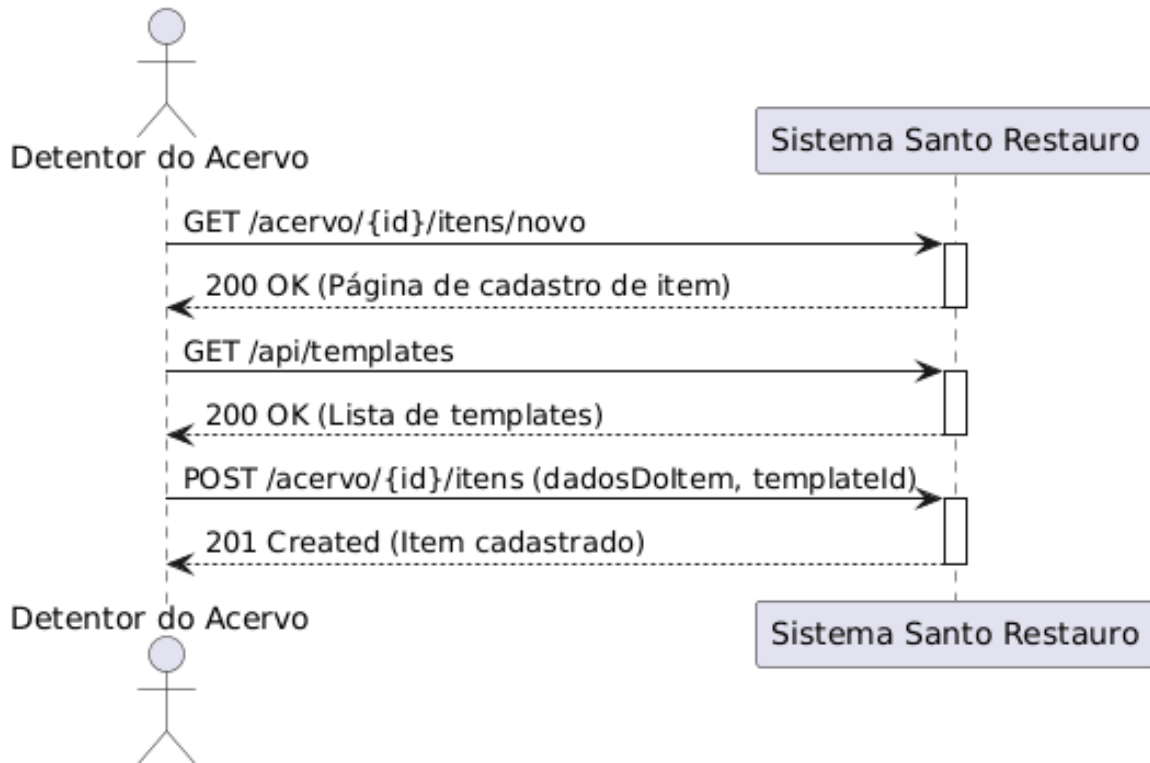
2.3.1 Diagrama de Sequência - Fazer Login

Este diagrama descreve o fluxo de eventos quando um Usuário (seja Detentor ou Administrador) se autentica na plataforma.



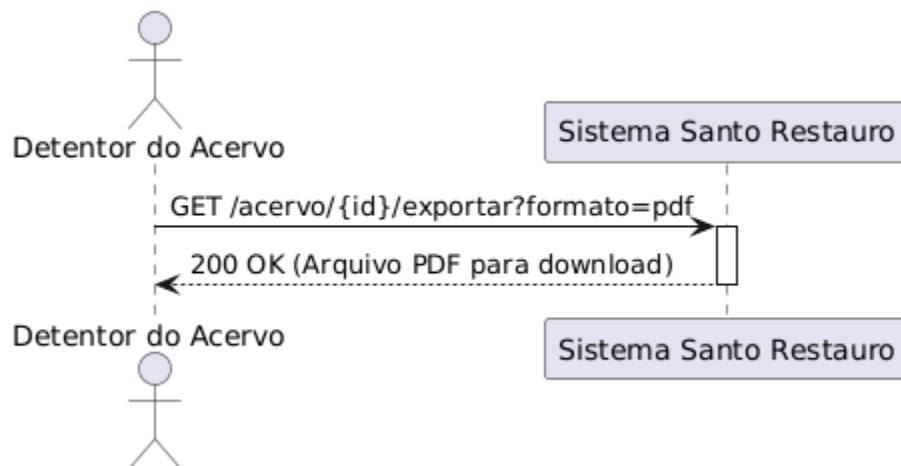
2.3.2 Diagrama de Sequência - Adicionar Item ao Acervo

Este diagrama descreve o fluxo principal do Detentor do Acervo ao cadastrar um novo item. Ele inclui os passos de solicitar o formulário, selecionar um template e enviar os dados.



2.3.3 Diagrama de Sequência - Exportar Acervo para PDF

Este diagrama descreve o fluxo opcional (<<extend>>) onde um Detentor do Acervo (ou Administrador) solicita a exportação dos dados do acervo em formato PDF.



2.3.4 Contratos de operação

Nesta subseção são detalhados os contratos para as principais operações do sistema identificadas nos Diagramas de Sequência do Sistema.

2.3.4.1

Contrato de Operação: Autenticação de Usuário

Contrato	CO-01: Autenticação de Usuário
Operação	POST /login(email, senha)
Referências cruzadas	UC: Fazer login
Pré-condições	1. O Usuário (Detentor ou Admin) não está autenticado. 2. O email e senha fornecidos não estão vazios.
Pós-condições	1. As credenciais (email, senha) foram validadas com sucesso. 2. Uma nova sessão foi criada e associada ao Usuário correspondente. 3. Um token de sessão (JWT) foi gerado e retornado ao Ator. 4. O sistema agora considera o Usuário como "autenticado".

2.3.4.2

Contrato de Operação: Adicionar Item ao Acervo

Contrato	CO-02: Adicionar Item ao Acervo
Operação	POST /acervo/{id}/itens (dadosDoItem, templateId)
Referências cruzadas	UC: Gerir itens de um acervo
Pré-condições	1. O Usuário (Detentor ou Admin) está autenticado. 2. O Usuário possui permissão para modificar o acervo com id informado. 3. O acervo/{id} existe no sistema. 4. O templateId existe e é válido. 5. Os dadosDoItem estão em conformidade com os campos do templateId.
Pós-condições	1. Uma nova instância de Item foi criada e persistida no banco de dados. 2. A instância de Item foi associada ao Acervo (via Acervo_fk). 3. A instância de Item foi associada ao Template (via Template_fk). 4. Os dadosDoItem (JSON) foram salvos na coluna dados_item do novo item. 5. O sistema registrou a data de criação do item.

2.3.4.3

Contrato de Operação: Exportar Acervo para PDF

Contrato	CO-03: Exportar Acervo para PDF
Operação	GET /acervo/{id}/exportar?formato=pdf
Referências cruzadas	UC: gerir seu acervo (via <<extend>>)

	UC: Gerir acervos (via <<extend>>)
Pré-condições	<ol style="list-style-type: none"> 1. O Usuário (Detentor ou Admin) está autenticado. 2. O Usuário possui permissão para visualizar o acervo/{id}. 3. O acervo/{id} existe no sistema. 4. O parâmetro formato=pdf é um formato de exportação válido.
Pós-condições	<ol style="list-style-type: none"> 1. O sistema consultou os dados do acervo/{id} e todos os Itens associados a ele. 2. Um arquivo binário (PDF) foi gerado em memória contendo os dados consultados. 3. O arquivo PDF foi retornado ao Ator (iniciando o download). 4. Importante: Nenhum estado do sistema (dados no banco) foi alterado.

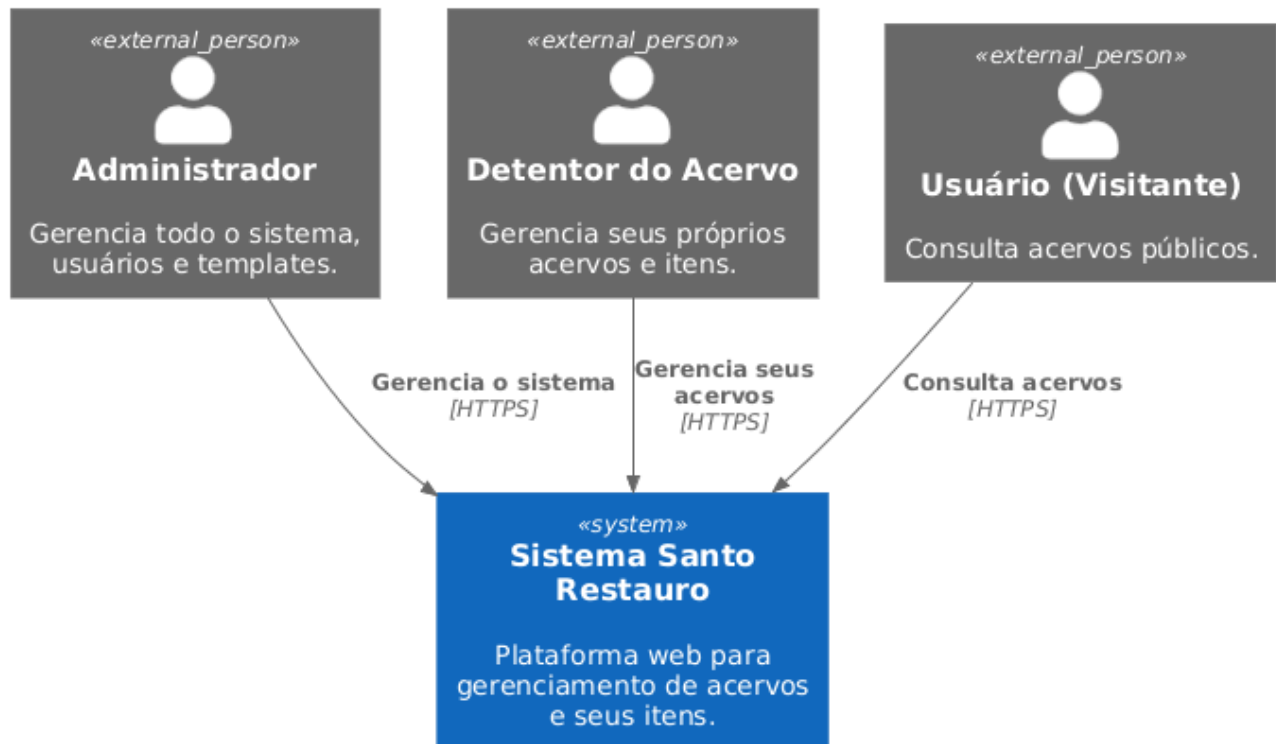
3. Modelos de Projeto

3.1 Arquitetura

A arquitetura do sistema Santo Restauo será descrita usando o C4 Model. O primeiro nível, o Diagrama de Contexto (C1), é exibido abaixo. Ele ilustra a visão de mais alto nível do sistema, mostrando-o como uma única "caixa preta", os atores que interagem com ele (definidos na Seção 2.1) e suas principais interações.

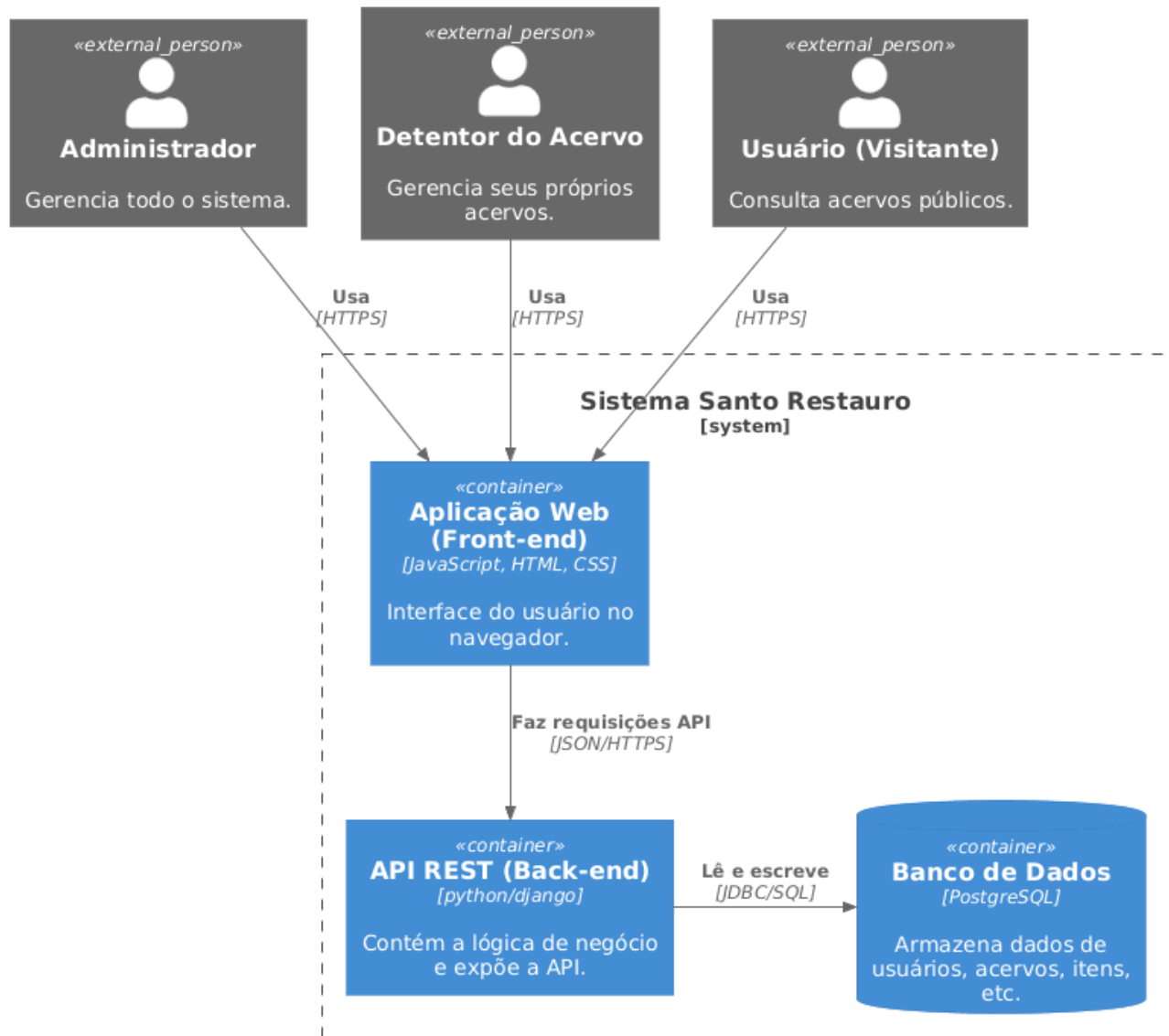
3.1.1

Nível 1: Diagrama de Contexto (C1)



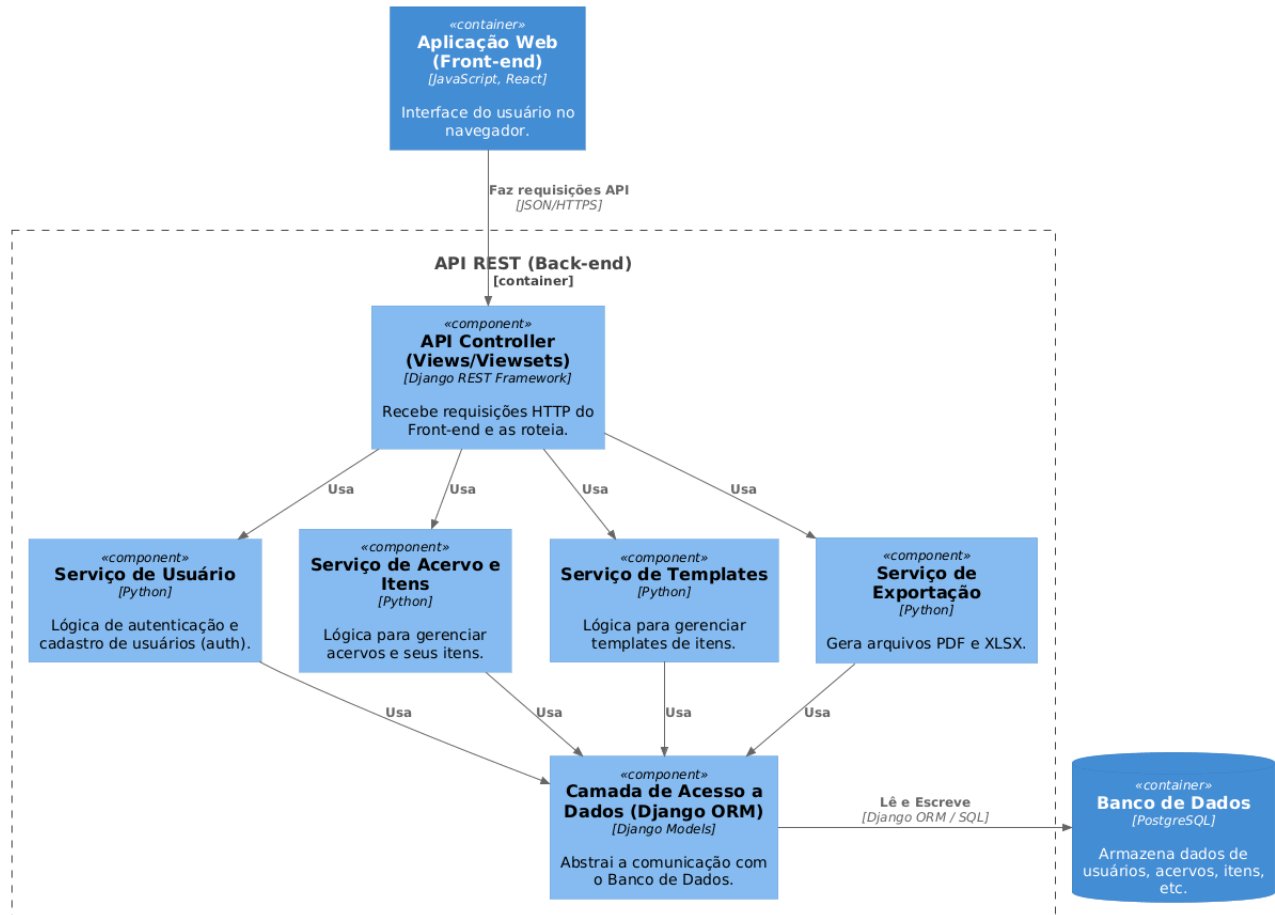
3.1.2

Nível 2: Diagrama de Contêineres (C2)

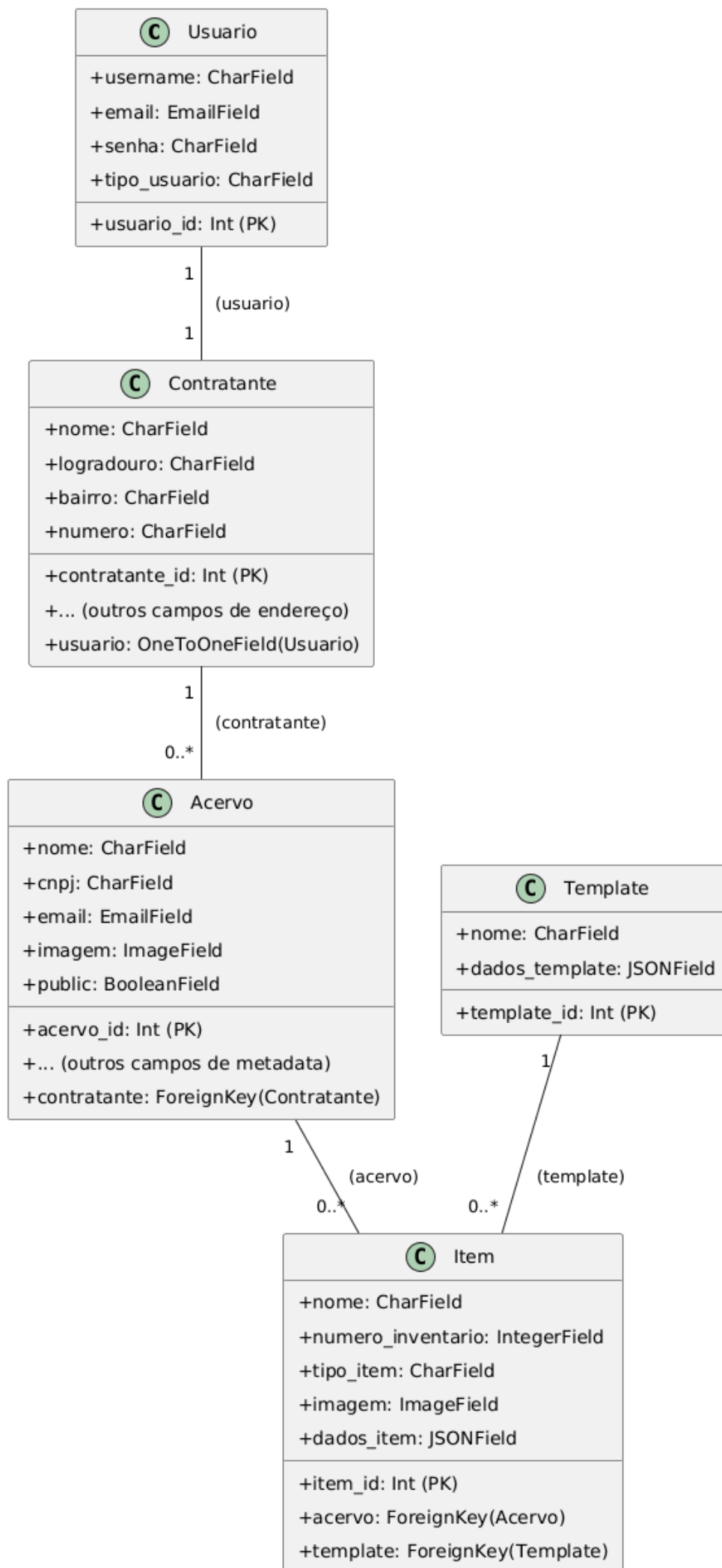


3.2 Diagrama de Componentes e Implantação.

Diagramas de componentes do sistema. Diagrama de implantação mostrando onde os componentes estarão alocados para a execução.



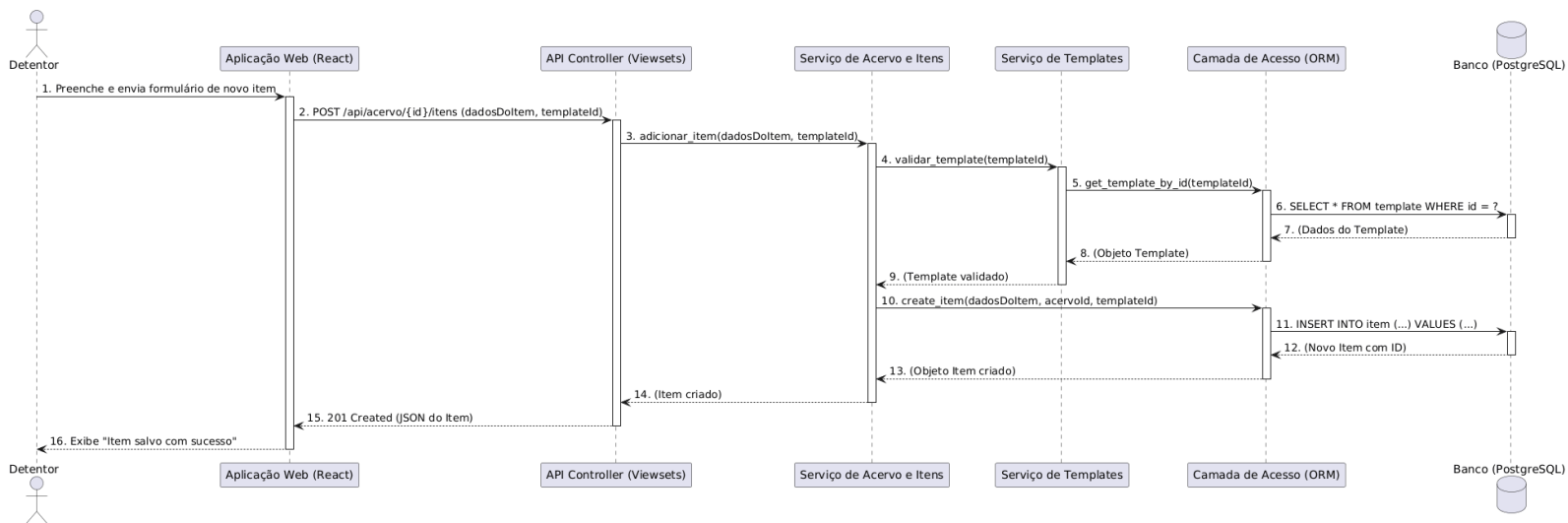
3.3 Diagrama de Classes



3.4 Diagramas de Sequência

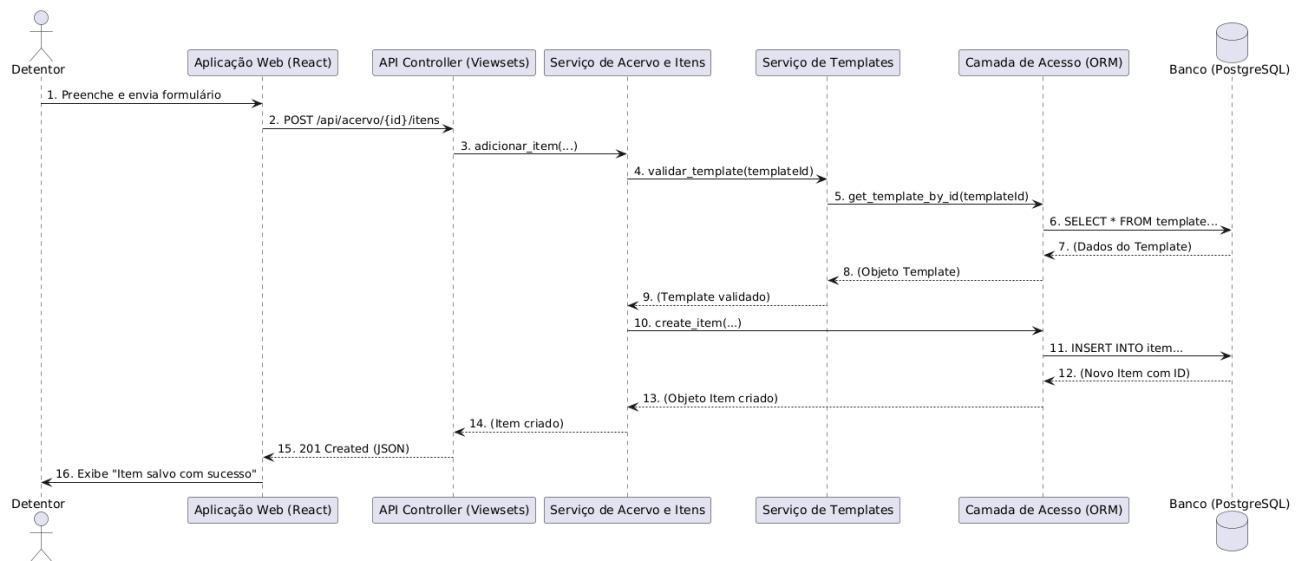
Os Diagramas de Sequência (ou Diagramas de Interação) a seguir detalham como os Componentes (definidos no C3, seção 3.2.1) colaboram internamente para realizar as operações do sistema.

O diagrama abaixo ilustra o fluxo de realização da operação **CO-02: Adicionar Item ao Acervo**.



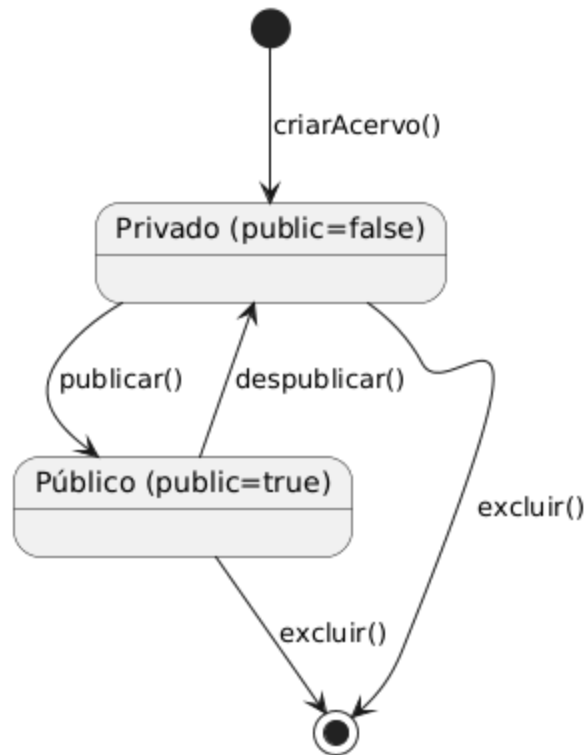
3.5 Diagramas de Comunicação

Diagramas de comunicação para realização da operação **CO-02: Adicionar Item ao Acervo**.



3.6 Diagramas de Estados

Ciclo de Vida do Acervo (Visibilidade)



4. Modelos de Dados

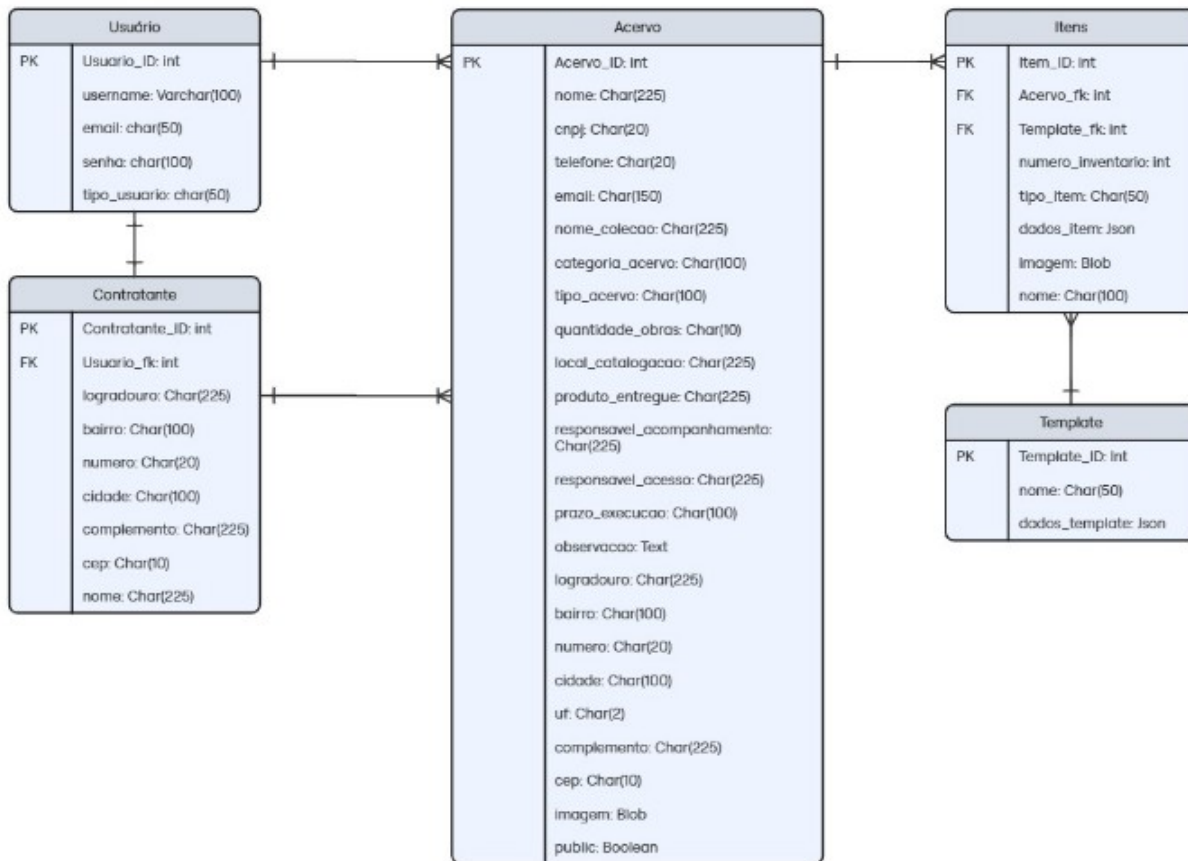
Nesta seção são apresentados os esquemas de banco de dados do sistema e as estratégias de mapeamento utilizadas para conectar a representação de objetos do código com o modelo de dados relacional.

4.1 Esquema de Banco de Dados

O esquema de banco de dados (ou Modelo Lógico) define a estrutura, as tabelas, as colunas e os relacionamentos do banco de dados PostgreSQL que dá suporte ao sistema.

Este esquema foi a base para a criação do Diagrama de Classes (Seção 3.4) e serve como a "fonte da verdade" para a persistência de dados.

Santo Restauro



4.2 Estratégias de Mapeamento (Objeto-Relacional)

A estratégia de mapeamento entre as "representações de objetos" (código) e "não-objetos" (tabelas) é implementada através de um Mapeador Objeto-Relacional (ORM).

Considerando que a API é desenvolvida em Django, a ferramenta utilizada para esta estratégia é o Django ORM.

O Django ORM abstrai a complexidade do SQL e permite ao desenvolvedor interagir com o banco de dados usando apenas classes Python (`models.py`). O ORM é responsável por "traduzir" as classes em tabelas e as operações de objetos (ex: `item.save()`) em comandos SQL (ex: `INSERT INTO...`).

O mapeamento segue a seguinte lógica:

Representação de Objeto (Python / Django)	Representação Não-Objeto (Relacional / PostgreSQL)
Classe (<code>class Acervo(models.Model)</code>)	Tabela (<code>CREATE TABLE acervo (...)</code>)
Atributo (<code>nome = models.CharField(...)</code>)	Coluna (<code>nome VARCHAR(...)</code>)
Instância (<code>meu_item = Item(...)</code>)	Linha ((linha na tabela 'item'))
<code>models.ForeignKey(Acervo)</code>	<code>INTEGER acervo_fk + FOREIGN KEY constraint</code>
<code>models.BooleanField(default=False)</code>	<code>BOOLEAN DEFAULT false</code> (Ex: campo public)
<code>models.JSONField()</code>	<code>JSONB</code> (tipo de dado nativo do PostgreSQL)