

---

**Vacinei**

---

**Vacinei**  
**Documento de Arquitetura de Software**

**1.0**

Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	

## Histórico da Revisão

Data	Versão	Descrição	Autor
24/09/2020	1.0	Criação do Documento	Dominic Rocha de Paulo

Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	

## Índice

1.	Introdução	4
1.1	Objetivo	4
1.2	Escopo	4
1.3	Definições, Acrônimos e Abreviações	4
1.4	Referências	4
1.5	Visão Geral	4
2.	Representação Arquitetural	4
3.	Restrições e Metas Arquiteturais	4
4.	Visão de Casos de Uso	5
5.	Visão Lógica	5
5.1	Visão Geral	5
5.2	Pacotes de Design Significativos do Ponto de Vista da Arquitetura	5
5.3	Realizações de Casos de Uso	5
6.	Visão de Processos	5
7.	Visualização da Implementação	5
8.	Visão da Implementação	5
8.1	Visão Geral	5
8.2	Camadas	6
9.	Visão de Dados (opcional)	6
10.	Tamanho e Desempenho	6
11.	Qualidade	6

## Documento de Arquitetura de Software

### 1. Introdução

Este documento apresenta a arquitetura proposta para um sistema de controle de vacinação, onde é utilizado várias técnicas para apresentar através de um conjunto de visões que juntas visam cobrir os principais aspectos técnicos relativos ao desenvolvimento e implantação do sistema em questão. O objetivo é capturar e formalizar as principais decisões tomadas com relação à arquitetura do sistema.

#### 1.1 Objetivo

Este documento tem por objetivo documentar a arquitetura de software, apresentando visões arquiteturais do sistema de gerenciamento de vacina, levando em consideração os atributos de qualidade escolhidos para o projeto.

Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	

## 1.2 Escopo

O escopo deste documento é apresentar várias visões diferentes para auxiliar no desenvolvimento do projeto orientando todos os envolvidos técnicos sobre o design de arquitetura escolhida para o sistema.

## 1.3 Definições, Acrônimos e Abreviações

[Esta subseção fornece as definições de todos os termos, os acrônimos e as abreviações necessários para a interpretação apropriada do **Documento de Arquitetura de Software**. Essas informações podem ser fornecidas em relação ao Glossário do projeto.]

## 1.4 Referências

[Esta subseção fornece uma lista completa de todos os documentos mencionados em outra parte do **Documento de Arquitetura de Software**. Identifique cada documento pelo seguinte: título, número do relatório (se for o caso), data e organização responsável pela publicação. Especifique as origens a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

## 1.5 Visão Geral

[Esta subseção descreve o que o restante do **Documento de Arquitetura de Software** contém e explica como o **Documento de Arquitetura de Software** é organizado.]

## 2. Representação Arquitetural

A arquitetura definida para o sistema é o padrão MVC na plataforma Web. A arquitetura foi definida com base no conhecimento dos envolvidos técnicos do projeto e segundo os atributos de qualidade escolhidos, como testabilidade, modularidade, modificabilidade e usabilidade.

Visão	Público	Área	Modelo da MDS
Lógica	Analistas	Realização dos Casos de Uso	
Processo	Integradores	Performance, Escalabilidade, Concorrência	
Implementação	Programadores	Componentes de Software	
Implantação	Gerência de Configuração	Nodos físicos	
Caso de Uso	Todos	Requisitos funcionais	
Dados	Especialistas em dados	Persistência de dados	

Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	

	Administradores de dados		
--	--------------------------	--	--

Visão de Caso de Uso

Apresenta funcionalidades importantes

Visão de Processos

Visão Lógica

Descreve as classes do projeto e seus relacionamentos apresentando a organização

Visão Componentes/Pacotes

### 3. Restrições e Metas Arquiteturais

- Estrutura MVC;
- Linguagem de programação PHP e JavaScript;
- Bancos de dados MySql;

Requisito	Solução
Linguagem	PHP e JavaScript
Plataforma	Web
Segurança	[Especificar a necessidade de segurança e as características básicas da segurança.]
Persistência	MySql
Internacionalização (i18n)	[Especificar a necessidade de internacionalização/localização na aplicação.]

### 4. Visão de Casos de Uso

[Esta seção lista os casos de uso ou cenários do modelo de casos de uso se eles representam alguma funcionalidade central significativa do sistema final ou se têm uma ampla cobertura arquitetural—se eles experimentam muitos elementos arquiteturais ou se enfatizam ou ilustram um ponto frágil específico da arquitetura.]

Nesta sessão está presente os casos de usos significativos para a arquitetura do sistema.

Esta seção lista as especificações centrais e significantes para a arquitetura do sistema, os itens em negritos desta seção são os casos de uso central.

Lista de casos de uso do sistema:

- Criar Conta
- Recuperar Conta
- Realizar Login
- Deletar Conta

Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	

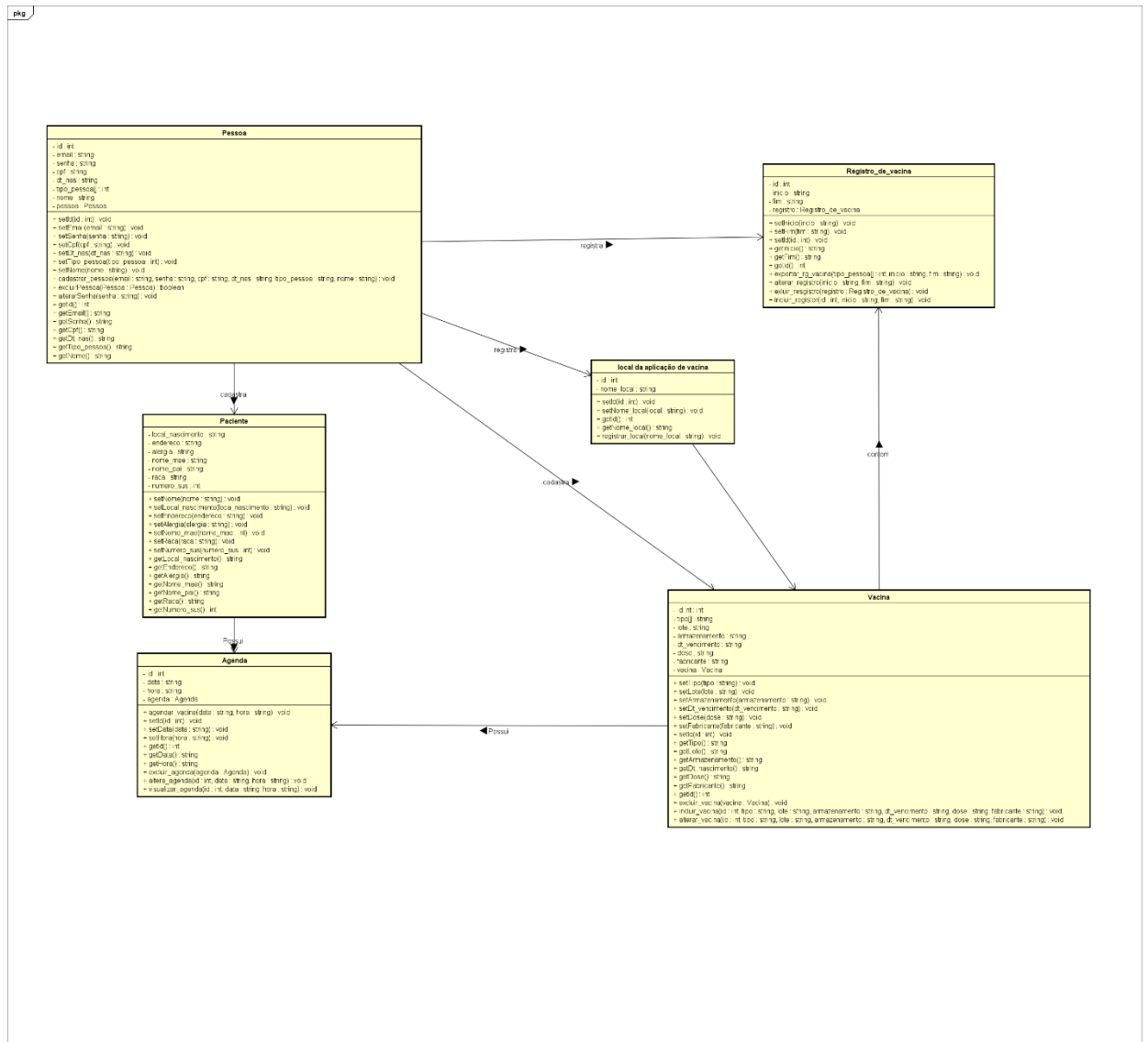
- **Exportar Registro das Vacinas**
- **Agendar Aplicação Vacina**
- Editar Registro de Aplicação de Vacina
- **Registrar Aplicação de Vacina**
- **Cadastrar Tipo de Vacina**
- **Visualizar Cartão de Vacina Virtual**
- Visualizar Vacinas Aplicadas
- Cadastrar local de aplicação de vacina
- Cadastrar tipo de vacina

<b>Nome do Caso de Uso</b>	<b>Agendar Aplicação Vacina</b>
<b>ID Caso de Uso</b>	<b>UC06</b>
<b>Descrição</b>	1. O usuário deve ser capaz de agendar a aplicação de uma vacina.
<b>Atores</b>	1. Usuário
<b>Pré-Condições</b>	1. O ator deverá estar logado. 2. Deve haver ao menos uma vacina cadastrada.
<b>Pós-Condições</b>	1. Vacina agendada.
<b>Fluxo Principal</b>	1. O ator seleciona a opção de agendamento. 2. O sistema direciona o ator para a página solicitada. 3. O ator seleciona a data e hora. 4. O sistema exibe tela de cadastro do paciente. 5. O ator informa os dados do paciente. 6. O sistema exibe o localizador de vacinas. 7. O ator seleciona o local de administração da vacina. 8. O ator seleciona os dados da vacina. 9. O sistema emite um alerta de agendamento realizado e muda a cor da data marcada.
<b>Fluxo Alternativo</b>	<b>A1 – Horário não disponível</b> 1. No passo 3 do fluxo principal o ator tenta marcar

Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	

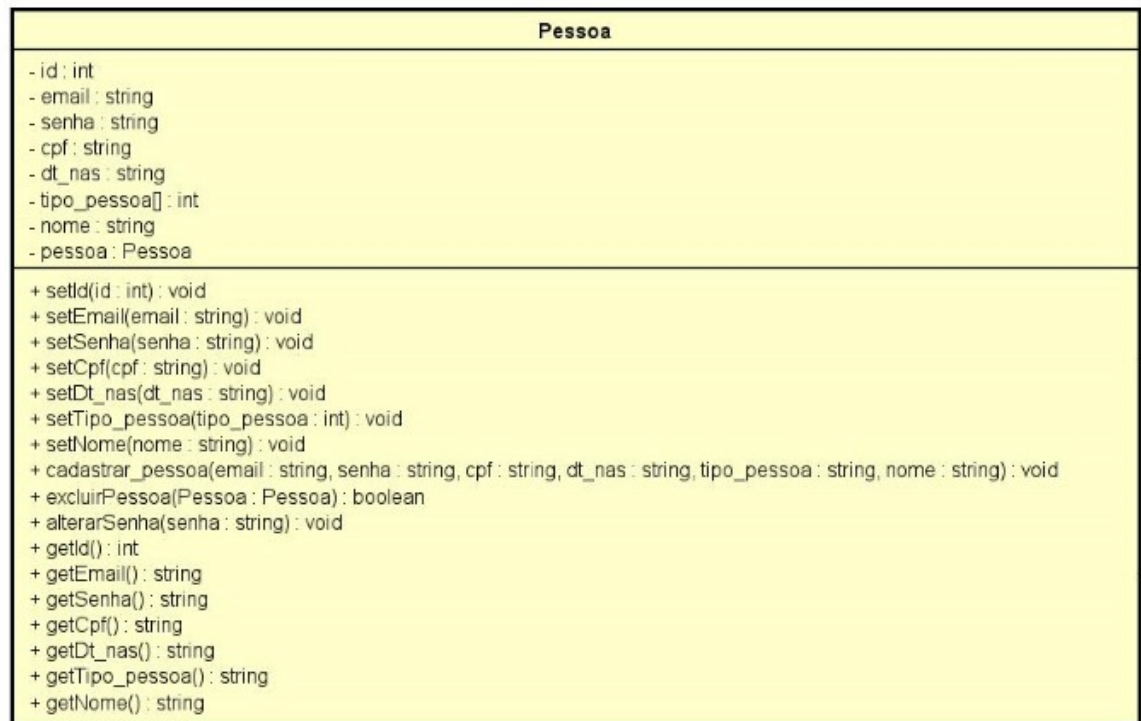
	<p>em um horário do dia que já está agendado.</p> <ol style="list-style-type: none"> <li>2. O sistema alerta o ator que já existe um paciente agendado para aquele horário e aborta a ação.</li> <li>3. Sistema retorna para o passo 3 do fluxo principal.</li> </ol> <p><b>A2 – Vacina não cadastrada</b></p> <ol style="list-style-type: none"> <li>1. No passo 3 do fluxo principal o ator não consegue localizar a vacina.</li> <li>2. O sistema alerta não encontrou nenhum resultado com o filtro selecionado.</li> <li>3. O sistema solicita se o ator deseja realizar o cadastro da vacina e direciona para o caso de uso UC09.</li> </ol>
	<b>Exceções</b>

## 5. Visão Lógica

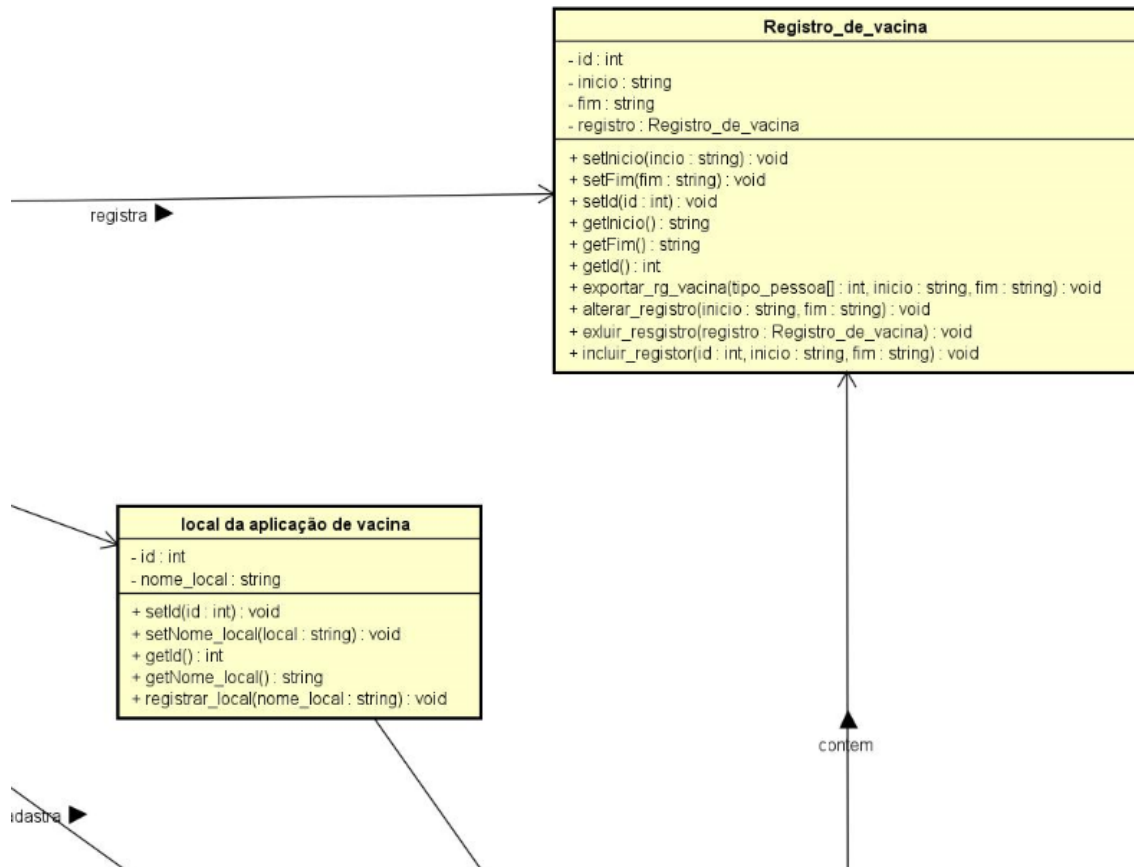




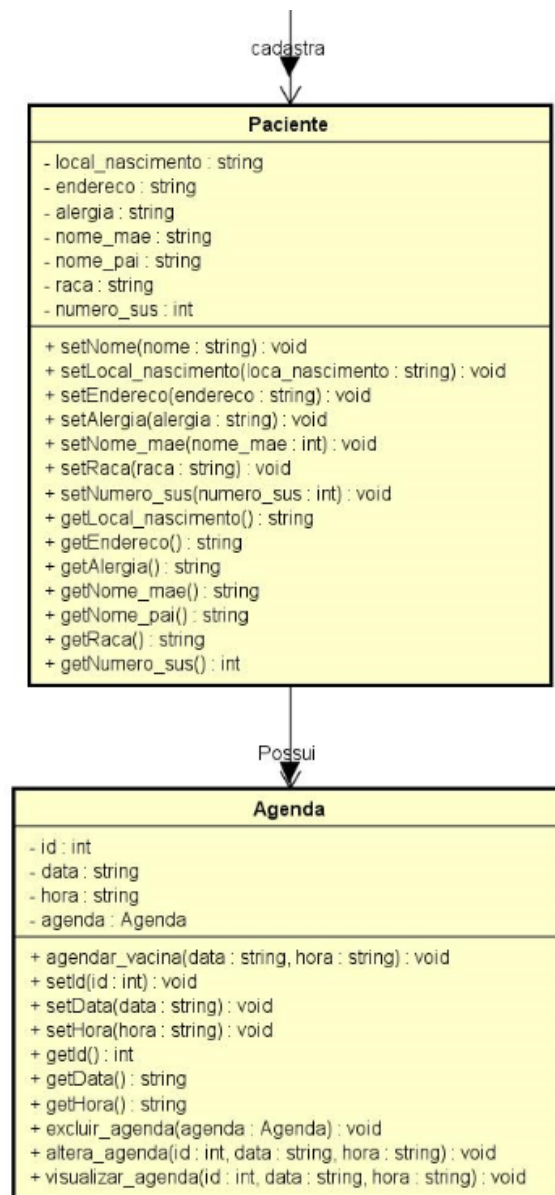
Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	



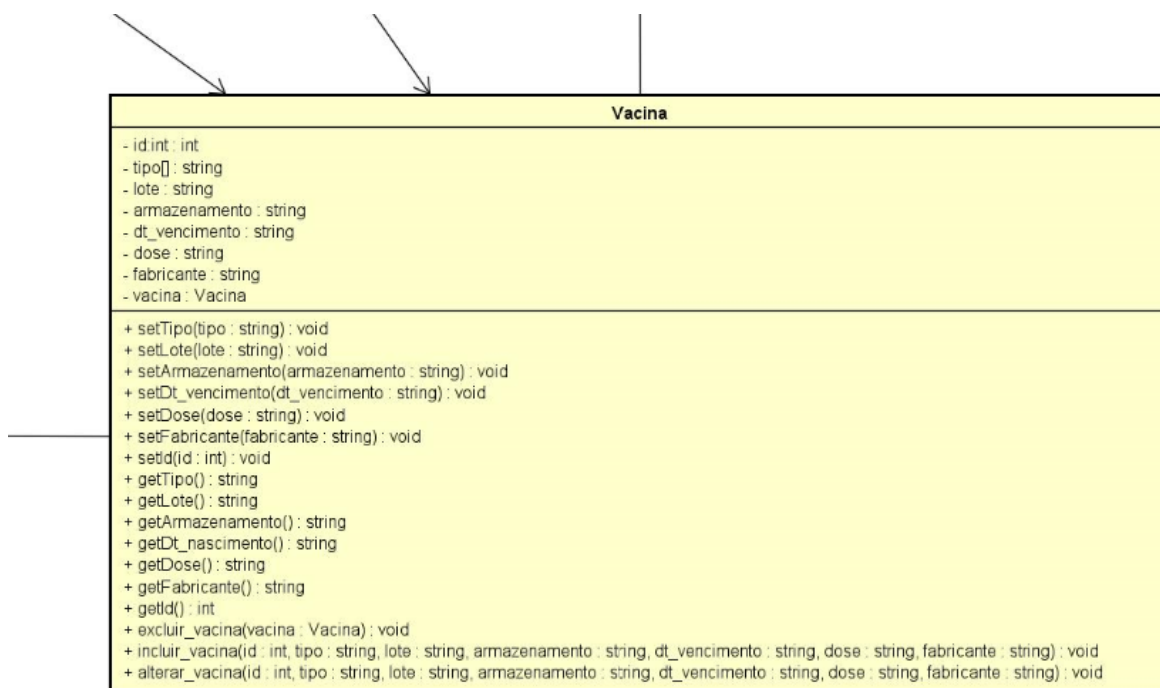
Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	



Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	



Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	



## 5.1 Visão Geral

[Esta subseção descreve toda a decomposição do modelo de design em termos de camadas e de hierarquia de pacotes.]

## 5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura

[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos.

Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]

## 5.3 Realizações de Casos de Uso

[Esta seção ilustra o funcionamento do software, apresentando algumas realizações (ou cenários) de casos de uso selecionadas e explica como os diversos elementos do modelo de design contribuem para a respectiva funcionalidade.]

## 6. Visão de Processos

[Esta seção descreve a decomposição do sistema em processos leves (encadeamentos simples de controle) e processos pesados (agrupamentos de processos leves). Organize a seção em grupos de processos que se comunicam ou interagem. Descreva os modos principais de comunicação entre processos, como transmissão de mensagens e interrupções.]

## 7. Visão da Implementação

[Esta seção descreve a estrutura geral do modelo de implementação, a divisão do software em camadas e subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da

Vacinei	Versão: 1.0
Documento de Arquitetura de Software	Data: 24/09/2020
<identificador do documento>	

arquitetura.]

### 7.1 Visão Geral

[Esta subseção nomeia e define as diversas camadas e o seu conteúdo, as regras que determinam a inclusão em uma camada específica e as fronteiras entre as camadas. Inclua um diagrama de componentes que mostre os relacionamentos entre as camadas. ]

### 7.2 Camadas

[Para cada camada, inclua uma subseção com o respectivo nome, uma lista dos subsistemas localizados na camada e um diagrama de componentes.]

## 8. Visão de Dados

[Uma descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou inexistentes ou se a conversão entre o Modelo de Design e o Modelo de Dados for trivial.]

## 9. Tamanho e Desempenho

[Uma descrição das principais características de dimensionamento do software que têm um impacto na arquitetura, bem como as restrições do desempenho desejado.]

## 10. Qualidade

[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: capacidade de extensão, credibilidade, portabilidade e assim por diante. Se essas características possuírem significado especial, como implicações de segurança, garantia ou privacidade, elas deverão ser delineadas claramente.]