MissM|Fotografia

Documento de Arquitetura de Software

Versão <2.0>

Índice Analítico

Conteúdo

[1. Introdução 4](#_Toc321036874)

[1.1 Finalidade 4](#_Toc321036875)

[1.2 Escopo 4](#_Toc321036876)

[1.3 Definições, Acrônimos e Abreviações 4](#_Toc321036877)

[1.4 Referências 4](#_Toc321036878)

[1.5 Visão Geral 4](#_Toc321036879)

[2. Representação Arquitetural 5](#_Toc321036880)

[3. Metas e Restrições da Arquitetura 5](#_Toc321036881)

[4. Visão de Casos de Uso 5](#_Toc321036882)

[4.1 Realizações de Casos de Uso 7](#_Toc321036883)

[5. Visão Lógica 8](#_Toc321036884)

[5.1 Visão Geral 8](#_Toc321036885)

[5.2 Pacotes des Design Significativos do Ponto de Vista da Arquitetura 9](#_Toc321036886)

[5.3 Diagrama de componentes 10](#_Toc321036887)

[6. Visão de Processos 10](#_Toc321036888)

[7. Visão de Implantação 11](#_Toc321036889)

[8. Visão da Implementação 11](#_Toc321036890)

[9. Visão de Dados (opcional) 11](#_Toc321036891)

[10. Tamanho e Desempenho 11](#_Toc321036892)

[11. Qualidade 11](#_Toc321036893)

[12. Exceções 11](#_Toc321036894)

Documento de Arquitetura de Software

# Introdução

Esse documento provê uma visão de alto nível dos objetivos da arquitetura, dos estilos arquiteturais e componentes que foram selecionados para poder implementar as funcionalidades propostas pelos casos de usos levantados do MissM|Fotografia.

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema MissM|Fotografia, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema de Linguagem Unificada de Modelagem (UML – *UnifiedModelingLanguage*).

## Escopo

Este Documento de Arquitetura de Software é aplicado ao MissM|Fotografia, que será desenvolvido pelos alunos Breno dos Reis Rezende, Tarcísio Oliveira e Victor Menezes do curso de Sistemas de Informação da Universidade Federal de Sergipe como projeto da disciplina Engenharia de Software II.

## Definições, Acrônimos e Abreviações

Todas as definições juntamente com os termos, acrônimos e abreviações necessárias à adequada interpretação e entendimento deste documento podem ser encontradas no documento de Glossário do Projeto.

## Referências

Os seguintes documentos foram utilizados como referência para a elaboração do documento arquitetura:

* Modelo de Análise
* Modelo de Regra de Negócio
* Modelo de Casos de Uso
* Descrição de Casos de Uso
* Documento de Requisitos Funcionais e Não Funcionais
* *RationalUnifiedProcess*

## Visão Geral

Com o objetivo de cobrir todos os aspectos da arquitetura, esse documento contém as seguintes subseções:

* Subseção 1: Descreve o uso de cada visão;
* Subseção 2: Descreve as restrições arquiteturais do sistema;
* Subseção 3: Descreve os requisitos funcionais que causam significante impacto na arquitetura;
* Subseção 4: Descreve a visão lógica da arquitetura;
* Subseção 5: Descreve a visão de processos;
* Subseção 6: Descreve a visão de implantação;
* Subseção 7: Descreve a visão de implementação;
* Subseção 8: Descreve a visão de dados;
* Subseção 9: Descreve as principais características de dimensionamento do software que têm um impacto na arquitetura;
* Subseção 10: Descreve como a arquitetura do software contribui para todos os recursos.
* Subseção 11: Mostra a hierarquia de exceções.

# Representação Arquitetural

Este documento apresenta a arquitetura como uma série de visualizações, mencionadas acima. Essas visões são apresentadas como Modelos do StarUML e utiliza a Linguagem Unificada de Modelagem (UML – *UnifiedModelingLanguage).*

Para representar a arquitetura do software foram utilizados como base os seguintes estilos arquiteturais:

* Camadas;
* Cliente-Servidor;
* Subrotinas;
* Baseado em Eventos;
* Repositório de banco de dados.

# Metas e Restrições da Arquitetura

Para que o software tenha um comportamento esperado pelos stakeholders ele deve as seguintes restrições:

* O sistema deve permitir a execução nos principais browsers disponíveis como por exemplo: *Mozilla Firefox, Chrome, Internet Explorer*;
* O sistema será implementado seguindo a arquitetura .NET, usando como linguagem o C#;
* O sistema terá os seus dados persistidos no banco de dados MySQL;
* Apenas usuários autenticados podem acessar o sistema.

# Visão de Casos de Uso

Nessa seção serão listados os casos de uso que estão representados no modelo de casos de uso. Esses casos de uso são:

* CSU01 –Autenticar Usuário;
* CSU02 – Manter Serviço;
* CSU03 – Manter Pagamento;
* CSU04 – Manter Evento;
* CSU05 – Gerar Relatório;
* CSU06 – Manter Tipos de Evento;
* CSU07 – Manter Fotos;
* CSU08 – Manter Cliente;
* CSU09 – Manter Orçamento;
* CSU10 – Manter Produto;
* CSU11 – Manter Contrato;
* CSU12 – Manter Agenda;
* CSU13 – Manter Funcionário;

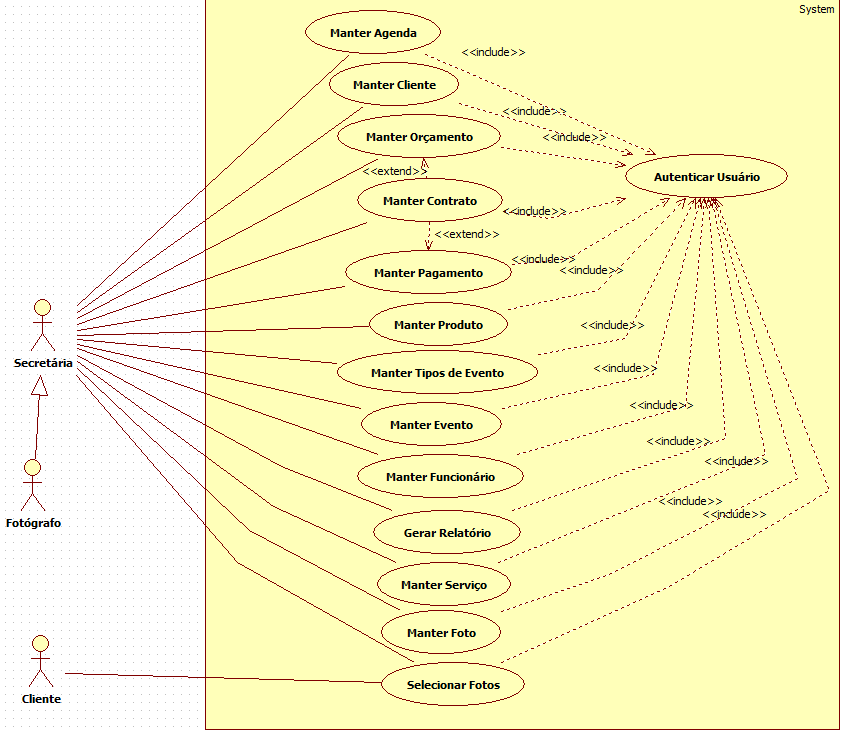


Figura 1 – Diagrama de Casos de Uso (Pacote Principal).

## Realizações de Casos de Uso

As descrições de cada caso de uso contido no diagrama da Figura 1 estão disponíveis na pasta ” Casos de uso”.

# Visão Lógica

## Visão Geral

A visão lógica do MissM|Fotografia é composta principalmente por quatro pacotes:

* ViewController: Nesse pacote são armazenados os componentes do tipo View, que fazem parte da interface gráfica da aplicação. Também serão armazenados os componentes do tipo Controller. Esses componentes são responsáveis por integrar os Models com as Views, de acordo com a interação com usuário;
* Service: Nesse pacote serão armazenadas as classes que representam as regras de negócio do sistema;
* Model: Nesse pacote serão armazenadas as classes que representam entidades da aplicação;
* Data: Nesse pacote serão armazenadas as classes de persistência da aplicação;

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

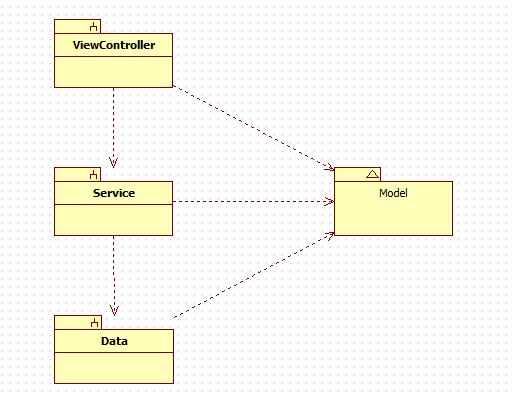


Figura 2 – Arquitetura

# Visão de Processos

Ainda não foi definida

# Visão de Implantação

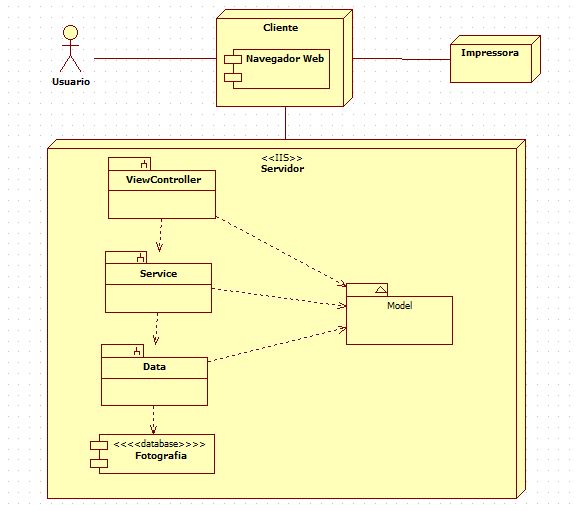


Figura 4 – Diagrama de Implantação

# Visão da Implementação

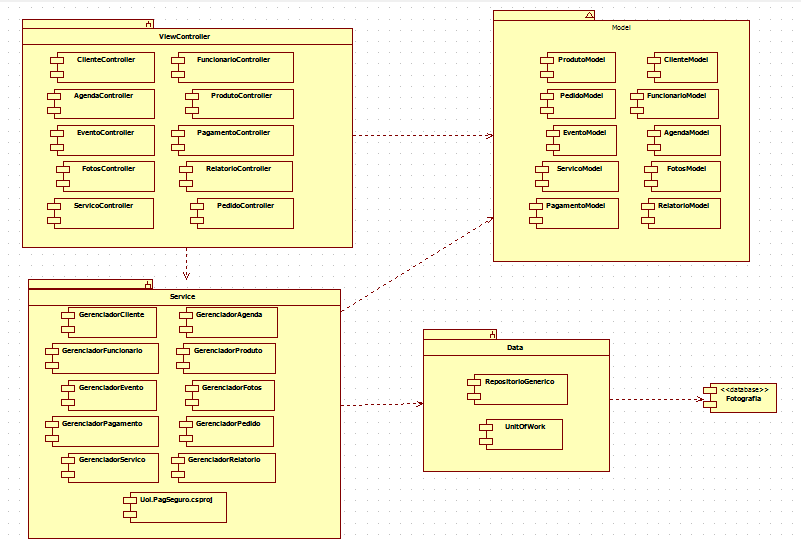
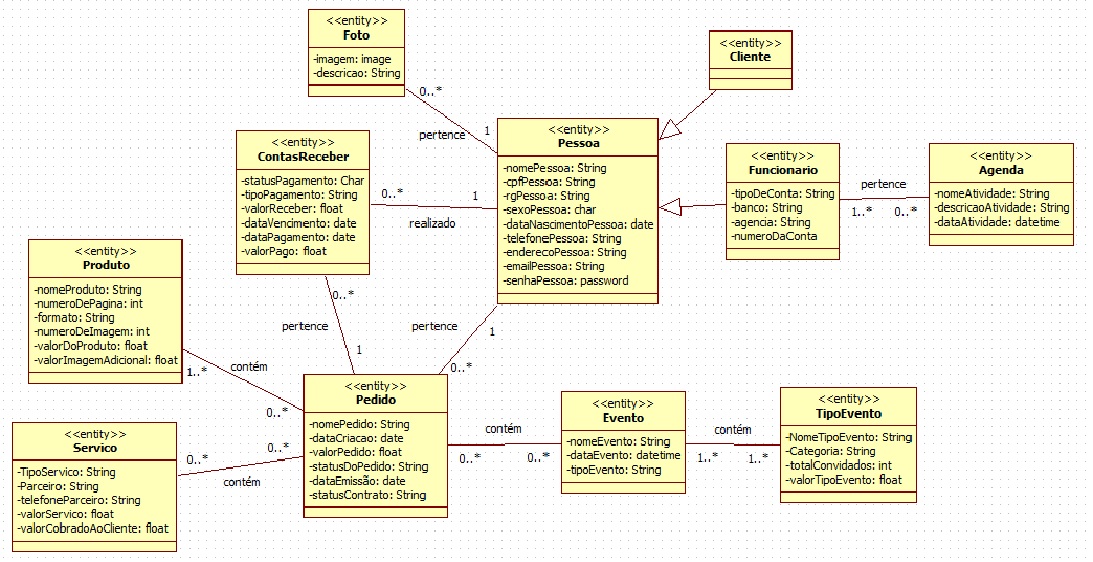


Figura 3 – Diagrama de Diagrama de Componentes

# Visão de Dados (opcional)



# Tamanho e Desempenho

O sistema armazenará imagens com tamanho entre 5Mb à 60Mb.

# Qualidade

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

# Exceções

O tratamento de exceções se dará em camadas. O sistema terá tratamento de exceções específicas definidas em *NegocioException, DataException e ModelException,* que utilizarão os tratamentos de exceções genéricos definidos em *System.Exception.*

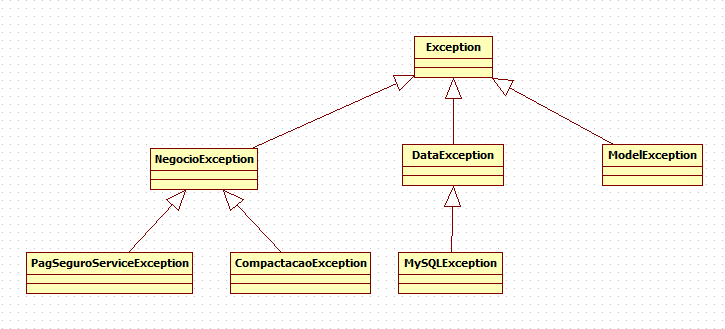


Figura 5 – Diagrama da Arquitetura de Exceções