FERRAMENTA PARA ELABORAR SURVEY EM ENGENHARIA DE SOFTWARE - MYSURVEY

Documento de Arquitetura de Software

Versão <2.0>

Índice Analítico

Conteúdo

[1. Introdução 3](#_Toc361651315)

[1.1 Finalidade 3](#_Toc361651316)

[1.2 Escopo 3](#_Toc361651317)

[1.3 Definições, Acrônimos e Abreviações 3](#_Toc361651318)

[1.4 Referências 3](#_Toc361651319)

[1.5 Visão Geral 3](#_Toc361651320)

[2. Representação Arquitetural 4](#_Toc361651321)

[3. Metas e Restrições da Arquitetura 4](#_Toc361651322)

[4. Visão de Casos de Uso 4](#_Toc361651323)

[4.1 Realizações de Casos de Uso 5](#_Toc361651324)

[5. Visão Lógica 5](#_Toc361651325)

[5.1 Visão Geral 5](#_Toc361651326)

[5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 6](#_Toc361651327)

[5.3 Diagrama de componentes 7](#_Toc361651328)

[6. Visão de Processos 8](#_Toc361651329)

[7. Visão de Implantação 8](#_Toc361651330)

[8. Visão da Implementação 8](#_Toc361651331)

[9. Tamanho e Desempenho 8](#_Toc361651333)

[10. Qualidade 8](#_Toc361651334)

[11. Exceções 9](#_Toc361651335)

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 estruturar as funcionalidades propostas pelos casos de usos da Ferramenta de Survey em Engenharia de Software (MYSURVEY).

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente da Ferramenta de Survey em Engenharia de Software (MYSURVEY), 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, utilizando a linguagem de modelagem unificada (UML – *Unified Modeling Language*).

## Escopo

Este Documento de Arquitetura de Software é aplicado a Ferramenta de Survey em Engenharia de Software (MYSURVEY), que será desenvolvido pelos alunos Erivan Lima Santana e Antonio Douglas Oliveira Cunha do curso de Sistemas de Informação da Universidade Federal de Sergipe – Campus Itabaiana, e o professor Msc. Marcos Barbosa Dósea do Departamento de Sistemas de Informação da Universidade Federal de Sergipe – Campus Itabaiana, como projeto da disciplina Trabalho de Conclusão de Curso II.

## Definições, Acrônimos e Abreviações

|  |  |
| --- | --- |
| Termo | Explicação |
| MySurvey | Ferramenta para criação de survey voltado a engenharia de software |
| Responsável | Usuário da ferramenta com privilégio para gerenciar seus surveys. |
| Entrevistado | Usuário que poderá responder os surveys criados por um Responsável. |
| Survey | Tipo de questionário aplicado na WEB |

## Referências

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

* 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
* *Rational Unified Process*

## Visão Geral

Com o objetivo de contemplar todos os aspectos da arquitetura, estratificamos o documento nas seguintes subseções:

* Subseção 2: Descreve o uso de cada visão;
* Subseção 3: Descreve as restrições arquiteturais do sistema;
* Subseção 4: Descreve os requisitos funcionais que causam significante impacto na arquitetura;
* Subseção 5: Descreve a visão lógica da arquitetura;
* Subseção 6: Descreve a visão de processos;
* Subseção 7: Descreve a visão de implantação;
* Subseção 8: Descreve a visão de implementação;
* Subseção 9: Descreve a visão de dados;
* Subseção 10: Descreve as principais características de dimensionamento do software que têm um impacto na arquitetura;
* Subseção 11: Descreve como a arquitetura do software contribui para todos os recursos.
* Subseção 12: Mostra a hierarquia de exceções.

# Representação Arquitetural

Este documento apresenta a arquitetura como um arranjo das visualizações, mencionadas acima. Essas visões são apresentadas como Modelos do StarUML e utiliza a Linguagem de Modelagem Unificada (UML – *Unified Modeling Language).*

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

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

# Metas e Restrições da Arquitetura

Para que o software tenha um comportamento esperado por nossos Stakeholders, definimos as seguintes restrições:

* O Sistema deverá ser *Cross-Browser*, ou seja, as funcionalidades por ele oferecidas deverão funcionar plenamente nos mais diversos browsers disponíveis no mercado, a exemplo, do *Mozilla Firefox*, *Google Chrome, Opera, Safari, Netscape* e *Internet Explorer*, em suas versões mais recentes;
* O sistema será implementado utilizando os recursos inicialmente disponibilizados pelo *.NET Framework,* e ASP.NET MVC, tendo como linguagem de programação principal o C#;
* O sistema terá os seus dados persistidos no banco de dados Microsoft SQL Server;
* Apenas usuários autenticados podem ser acessar o sistema;
* O Sistema será multiplataforma, funcionado principalmente nos sistemas operacionais das plataformas Windows, Linux e MAC OS, em suas versões mais recentes, principalmente;
* O Sistema terá como *Design Pattern* o modelo MVC.

# Visão de Casos de Uso

Nessa seção serão listados os casos de uso responsáveis por nos fornecer num alto-nível de abstração as principais funcionalidades e comportamentos esperados do sistema. Esses casos de uso são:

* CSU01- Autenticação;
* CSU02- Gerenciar *Survey*;
* CSU03- Manter Questões;
* CSU04- Enviar *Survey*;
* CSU05- Analisar Resultados;

## Realizações de Casos de Uso

A seguir temos uma breve descrição de cada caso de uso, utilizado na descrição das principais funcionalidades do sistema. São eles:

* CSU01- Autenticação – O Usuário da ferramenta poderá autenticar no sistema para usar funcionalidades disponíveis para o seu perfil;
* CSU02- Gerenciar Survey – O Responsável poderá Criar Novo, Editar e Excluir um survey;
* CSU03- Manter Questões – O usuário Responsável poderá Adicionar, Editar e Excluir questões de um survey;
* CSU04- Enviar Survey – O usuário Responsável poderá enviar um survey para email e rede social;
* CSU05- Analisar Resultados – O usuário Responsável poderá Gerar Relatório e Visualizar respostas do survey;

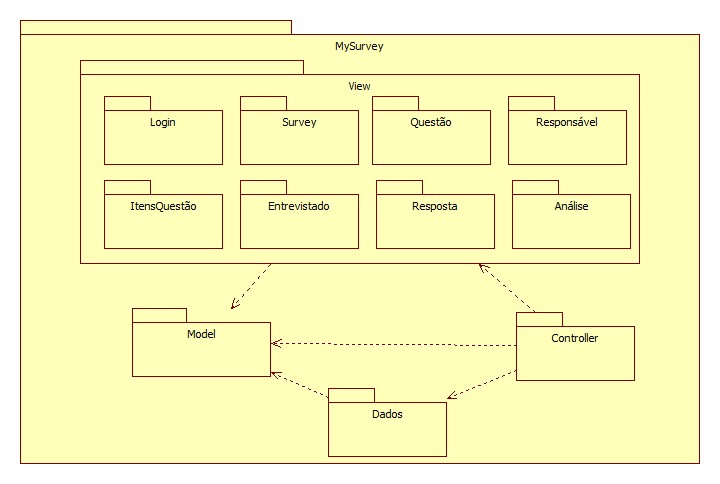
# Visão Lógica

## Visão Geral

A visão lógica do MYSURVEY é composta principalmente por três pacotes:

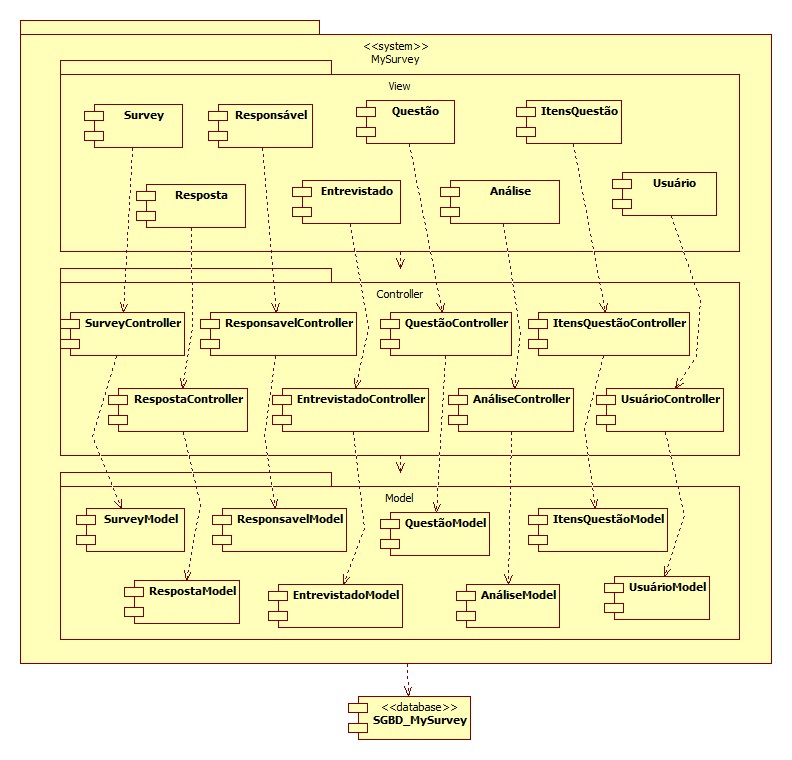
* Model: Esse pacote representa a implementação da parte lógica do domínio da aplicação. Aqui serão armazenadas classes que representam as entidades da aplicação e classes responsáveis pela persistência dos dados;
* Views: Nesse pacote são armazenados os componentes responsáveis por prover a interface gráfica da aplicação;
* Controller: Nesse pacote são armazenados os componentes do tipo Controller. Tais componentes são responsáveis por relacionar os Models às Views, de acordo com a interação do usuário.

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



**Figura 9 – Diagrama de Pacotes**

## Diagrama de componentes

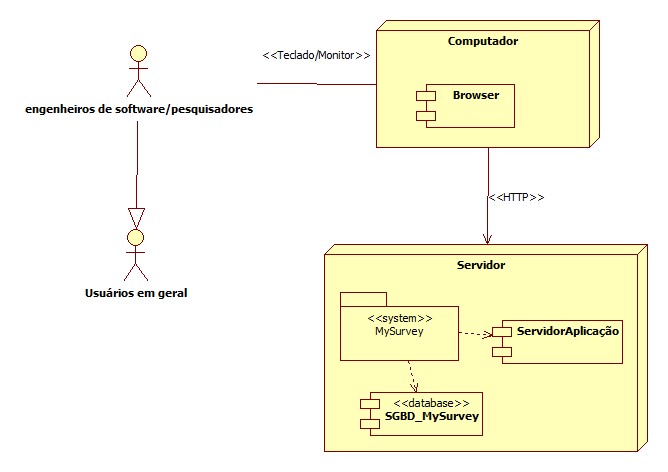


**Figura 10 – Diagrama de Componentes**

# Visão de Processos

Ainda não foi definida

# Visão de Implantação



**Figura 11 – Diagrama de Implantação**

# Visão da Implementação

A visão de implementação encontra-se detalhada no documento “Guia de Análise e Projeto”.

# Tamanho e Desempenho

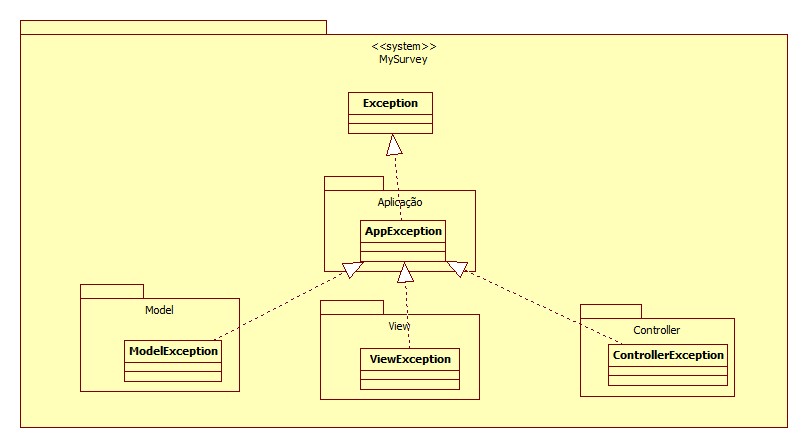
[Uma descrição das principais características de dimensionamento do software que têm um impacto na arquitetura, bem como as restrições do desempenho desejado.]

# 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 preferencialmente em camadas. O sistema possuirá tratamento de exceções específicas definidas pela *Application.Exception* que utilizará os tratamentos de exceções genéricos definidos em *System.Exception.*



**Figura 12 – Diagrama da Arquitetura de Exceções**