sJob

Documento de Arquitetura de Software

Versão 1.0

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 11/ago/17 | 1.0 | Detalhamento da arquitetura do sistema. | Vanessa e Yuri |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice Analítico

1. Introdução 4

1.1 Finalidade 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. Metas e Restrições da Arquitetura 5

4. Visão de Casos de Uso 5

4.1 Realizações de Casos de Uso 5

5. Visão Lógica 6

5.1 Visão Geral 6

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 6

5.3 Diagrama de Componentes 7

6. Visão de Processos 7

7. Visão de Implantação 8

8. Visão da Implementação 8

9. Qualidade 9

10. Exceções 9

Documento de Arquitetura de Software

# Introdução

Este documento traz em uma visão geral os objetivos da arquitetura, dos estilos arquiteturais e componentes que foram selecionados para poder estruturar as funcionalidades propostas pelos casos de usos levantados do sJob.

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema, 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.

## Escopo

Este documento de Arquitetura de Software está relacionado ao Sistema sJob, que será desenvolvido pelos alunos Vanessa Lima Santos e Yuri Santana Carvalho do curso de Sistemas de Informação da Universidade Federal de Sergipe – Campus Itabaiana, como projeto da disciplina Engenharia de Software 2.

## Definições, Acrônimos e Abreviações

Diagrama: Modelo gráfico.

UC: Use Case.

UML: *Unified Modeling Language*

## Referências

Para a criação do documento arquitetural, os documentos abaixo foram usados como referência:

* Diagrama de Casos de Uso
* Diagrama de Classe
* Diagrama de Pacotes
* Diagrama de Componentes
* Diagrama de Implantação

## Visão Geral

Para abordar todos os aspectos da arquitetura, o documento foi dividido nas seguintes subseções:

* + Subseção 2: Representação da arquitetura do sistema. Detalha cada visão usada;
  + Subseção 3: Descreve objetivos e restrições da arquitetura do sistema;
  + Subseção 4: Visão de Caso de Uso. Representação da funcionalidade;
  + Subseção 5: Detalhamento da visão lógica da arquitetura;
  + Subseção 6: Descreve a visão de processos;
  + Subseção 7: Detalha a visão de implantação;
  + Subseção 8: Descreve a visão de implementação;
  + Subseção 9: Descreve como a arquitetura do software contribui para todos os recursos;
  + Subseção 10: Mostra a hierarquia de exceções.

# Representação Arquitetural

A partir daqui, o documento irá apresentar de maneira mais detalhada os aspectos da arquitetura comentadas acima. As visões são apresentadas como Modelos do StarUML, utilizando a Linguagem de Modelagem Unificada (UML – *Unified Modeling Language*). A arquitetura do software é representada tendo como base alguns estilos arquiteturais:

* Camadas;
* Repositório de Banco de Dados.

# Metas e Restrições da Arquitetura

Para que a aplicação atinja os objetivos, algumas restrições foram criadas:

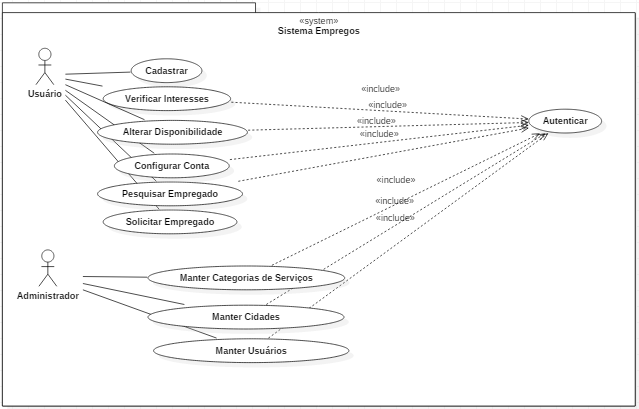
* O Sistema será implementado utilizando os recursos inicialmente disponibilizados pelo *.NET Framework* e ASP.NET MVC, tendo como linguagem de programação principal o C#;
* O sistema terá os seus dados persistidos no banco de dados PostgreSQL;
* Apenas usuários autenticados podem ser acessar o sistema;
* O Sistema terá como *Design Pattern* o modelo MVC.

# Visão de Casos de Uso

Nessa seção será apresentado o Diagrama de Caso de Uso que inclui os casos de uso que representam as funcionalidades do Sistema.

São eles:

* UC-003 - Usuário verifica interesses;
* UC-004 - Usuário altera disponibilidade;
* UC-006 - Usuário pesquisa empregado.



**Figura 1 – Diagrama de Casos de Uso**

## Realizações de Casos de Uso

Agora são descritos os casos de uso que estão associados ao funcionamento da aplicação.

São eles:

* UC-003 - Usuário verifica interesses - Verificar interesses;
* UC-004 - Usuário altera disponibilidade - Alterar disponibilidade;
* UC-006 - Usuário pesquisa empregado - Pesquisar empregados.

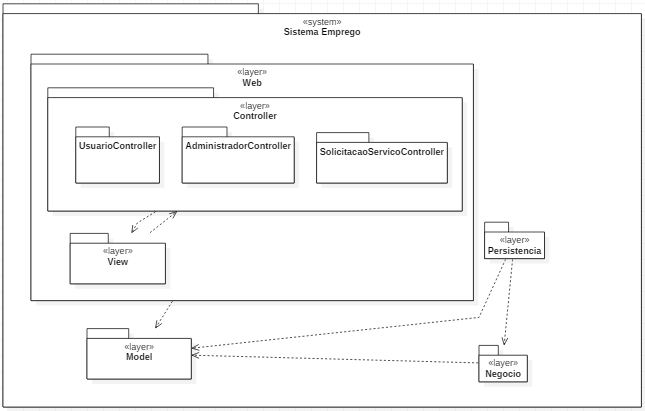
# Visão Lógica

## Visão Geral

A visão lógica do Sistema nJob é composta principalmente por 3 pacotes:

* Model: Representação do código programável, onde serão encontradas classes que representam as entidades do Sistema;
* View: Armazena componentes da interface gráfica do Sistema;
* Controller: Onde estão os componentes do tipo Controller, que são encarregados por relacionar os Models às Views.

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



**Figura 2 – Diagrama de Pacotes**

Sistema Emprego:

1. Web: Plataforma
2. Model: Codificação.
3. Persistencia: Armazenamento de dados.
4. Negocio: Regras de negócio.

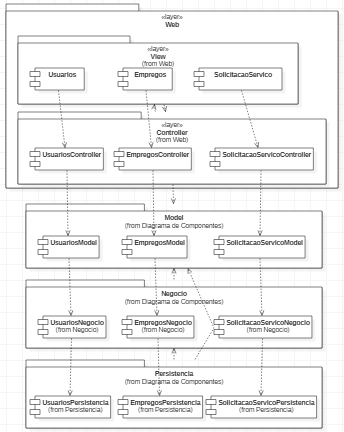
Web:

1. Controller: União Model-View.
2. View: Componentes gráficos.

Controller:

1. UsuarioController: Relacionamento Model - View, de acordo com a interação do usuário.
2. AdministradorController: Model - View, de acordo com a interação do usuário (administrador).
3. SolicitacaoServicoController: Model - View, de acordo com a interação do usuário.

## Diagrama de Componentes

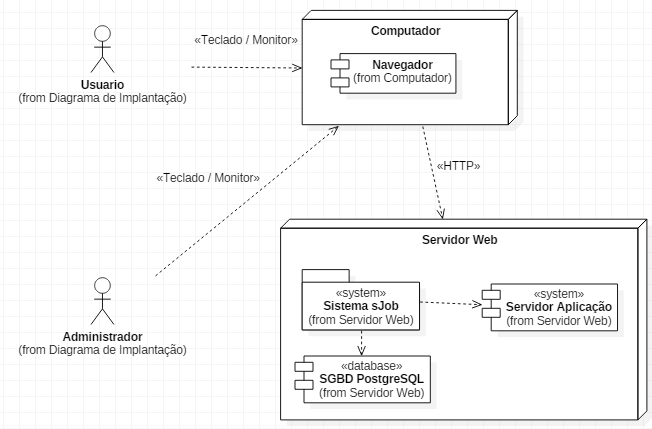


**Figura 3 – Diagrama de Componentes**

# Visão de Processos

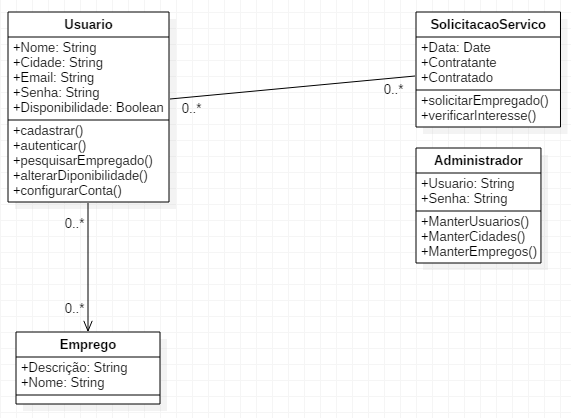
Não definido.

# Visão de Implantação



**Figura 4 – Diagrama de Implantação**

# Visão da Implementação



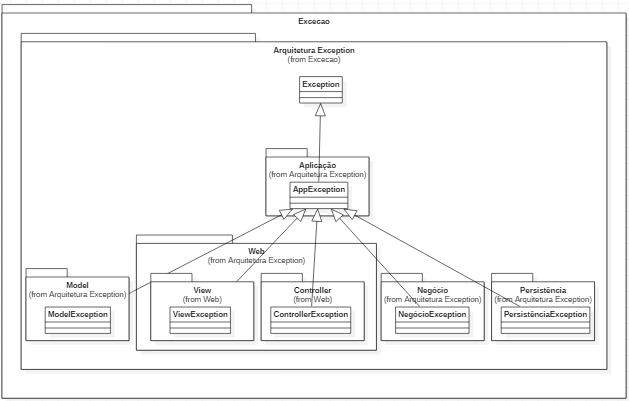
**Figura 5 – Diagrama de Classes**

# Qualidade

Requisitos não funcionais do Sistema.

# 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 6 – Diagrama da Arquitetura de Exceções**