# Faculdade de Juazeiro do Norte - FJN

Sistema de Controle de Estoque

Documentação Arquitetural

**Equipe:**

Ananda Rafaele Pinheiro Silva

Andressa Alves de Oliveira Souza

Cícero Tiago Galdino Bento Júnior

Jonas dos Santos Aquino

Jhonatan Bezerra Ferreira

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 20/05/2018 | 0.1 | Inicialização da Introdução | Cicero Tiago |
| 21/05/2018 | 0.2 | Inicialização da Representação Arquitetural | Jhonatan Bezerra |
| 24/05/2018 | 0.3 | Revisão do documento e aplicações de correção | Jhonatan Bezerra |
| 16/06/2018 | 0.4 | Definição dos requisitos e restrições arquiteturais | Cicero Tiago |
| 17/06/2018 | 0.5 | Definição dos requisistos de qualidade | Cicero Tiago |
| 17/06/2018 | 0.6 | Visão Lógica | Jhonatan Bezerra e Jonas Aquino |
| 17/06/2018 | 0.7 | Revisão geral da documentação | Cicero Tiago |

**Sumário**

[1. Introdução 4](#_Toc517189455)

[1.1 Finalidade 4](#_Toc517189456)

[1.2 Escopo 4](#_Toc517189457)

[1.3 Definições, Acrônimos e Abreviações 5](#_Toc517189458)

[1.4 Visão Geral 7](#_Toc517189459)

[2. Representação Arquitetural 7](#_Toc517189460)

[3. Requisitos e Restrições arquiteturais 8](#_Toc517189461)

[4. Qualidade 9](#_Toc517189462)

[5. Visão lógica 10](#_Toc517189463)

[6. Visão de implantação 13](#_Toc517189464)

# Introdução

O documento apresenta diversas perspectivas fundamentais da arquitetura proposta para o desenvolvimento do software SysEstoque. A arquitetura de software proposta conta com diversos padrões de projeto, principalmente com padrões Cliente-Servidor.

## Finalidade

O documento da arquitetura do software SysEstoque – Software de Controle de Estoque, objetiva fornecer de forma abrangente a arquitetura do mesmo, fazendo o uso de várias visões arquiteturais, do mesmo modo que proporcione a comunicação das decisões tomadas a respeito da arquitetura do software, enfatizando detalhes da macro arquitetura e requisitos não funcionais que impactem no mesmo.

Portanto, essa documentação é destinada para os desenvolvedores do software, analista de teste, analista de requisitos, gerente do projeto e para os interessados em analisar a estrutura arquitetural do software. Sua intenção é capturar e transmitir as decisões significativas do ponto de vista da arquitetura que foram tomadas em relação ao sistema.

## Escopo

Esse Documento de Arquitetura de Software – DAS concentra-se a documentar a arquitetura do sistema SysEstoque - Software de Controle de Estoque, que será desenvolvido pela equipe: Ananda Rafaele, Andressa Alves, Cicero Tiago, Jhonatan Bezerra, Jonas Aquino, onde contempla padrões de software, software de desenvolvimento, plataforma de desenvolvimento, componentes de software e frameworksque foram utilizados.

O presente documento contempla também os compostos, camadas que formam a arquitetura do sistema, além de demonstrar de forma visual tais compostos, com o intuito de auxiliar os programadores no momento de desenvolvimento do software.

Ainda presente no escopo, tal documento orienta todos os envolvidos técnicos da equipe de desenvolvimento do software, sendo ofertado orientações quanto as tecnologias utilizadas no desenvolvimento desse software.

## Definições, Acrônimos e Abreviações

As definições, acrônimos e abreviações utilizadas nesse documento estão listado no quadro abaixo com as respectivas descrições para melhor facilitar o entendimento e assimilação do presente documento.

|  |  |
| --- | --- |
| Aplicação | É um programa de computador com a funcionalidade de auxiliar os usuários nas suas atividades de acordo com cada atividade específica. |
| Software | Conjunto de componentes lógicos de um computador ou sistema de processamento de dados. |
| Hardware | Componentes físicos de um computador. |
| DAS | Documento de Arquitetura de Software. |
| Plataforma de desenvolvimento | É o ambiente na qual é construído o software. |
| Sistema Operacional | Programa ou um conjunto de programas cuja função é gerenciar os recursos do Sistema. |
| Framework | Abstração que une códigos comuns entre vários projetos de software. |
| Caso de Uso | Descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. |
| Diagrama | Representação gráfica de algo. |
| Desenvolvedor | Profissional que utiliza a linguagem de programação (PHP, Java, C etc.) para desenvolver programas de computador. |
| Escopo | É a soma de todos os componentes. |
| MVC | Padrão arquitetural de desenvolvimento baseado em camadas. |
| REST | **Transferência de Estado Representacional, é um estilo de arquitetura que define um conjunto de restrições e propriedades baseados em HTTP.** |
| Cliente/Servidor | Arquitetura na qual o processamento da informação é dividido em módulos ou processos distintos. Um processo é responsável pela manutenção da informação (servidores) e outros responsáveis pela obtenção dos dados (os clientes). |
| API | Toda parte do backend do SysEstoque. |
| Node.js | Interpretador JavaScript do lado do servidor. |
| MongoDB | Banco de dados não relacional estruturado em documentos. |
| Sails | Framework de arquitetura MVC para Node.js. |
| Bootstrap | O Bootstrap é um kit de ferramentas de código aberto para desenvolvimento com HTML, CSS e JS. |
| JQuery | Biblioteca de funções JavaScript que interage com o HTML, desenvolvida para simplificar os scripts interpretados no navegador do cliente. |
| Constraints | Restrições |
| Server-side | Termo usado para designar operações que são feitas no servidor. |

**Quadro 1: Definições, Acrônimos e Abreviações**

## Visão Geral

Este Documento de Arquitetura de Software possui aplicabilidade no desenvolvimento do software SysEstoque, no qual será desenvolvido pela fábrica de software Origami Desenvolvimento. De tal forma, que o documento auxilie na compreensão e no desenvolvimento da aplicação, onde será apresentada visões, que abordam a melhor escolha de arquitetura, que atenda as necessidades e resolva o problema do projeto.

# Representação Arquitetural

Tendo como base o problema a ser resolvido, o padrão arquitetural Model View Controller (MVC) foi a escolha de padrão para realizar o desenvolvimento do software SysEstoque. MVC separa a aplicação em 3 componentes. O de interação do usuário com o sistema (view), a de manipulação dos dados (model) e a de controle (controller).

Model é responsável pela leitura, escrita, atualização, exclusão dos dados, e também por efetuar as devidas validações relacionadas aos dados presente na base de dados.

View é o intermediário que proporciona o usuário interagir com o sistema, além de fornecer visualmente os dados.

Controller é responsável por receber todas as requisições do usuário. Os métodos que chamamos de actions são responsáveis por uma página, controlando qual model usar e qual view será mostrado ao usuário.

Visando uma programação orientada a protótipo, a complexidade da aplicação desenvolvida aumenta e torna-se adequado a separação entre os dados e a apresentação dos mesmos. Onde ao utilizarmos o padrão arquitetural MVC as alterações feitas na interface não afetam a manipulação dos dados, e estes poderão ser reorganizados sem alterar a interface, consequentemente aumentando o desacoplamento entre as partes.

Para tornar o SysEstoque coeso e performático decidimos utilizar o estilo arquitetural do Representational State Transfer (REST), em português Transferência de Estado Representacional, é um estilo de arquitetura que define um conjunto de restrições e propriedades baseados no protocolo HTTP. REST consiste em princípios/regras/constraints que, quando seguidas, permitem a criação de um projeto com interfaces bem definidas.

Para atender a necessidade do sistema SysEstoque ser executado em várias maquinas por usuários diferentes, decidimos utilizar a arquitetura Cliente/Servidor, em que o processamento de informações é dividido em processos distintos. Onde um processo é responsável pela manutenção da informação (Servidor) e o outro é responsável pela obtenção dos dados (Cliente), tornando os recursos centralizados. Funcionando da seguinte forma, os processos cliente enviam pedidos/requisições para o processo servidor, e este por sua vez processa e envia os resultados/respostas dos pedidos.

# Requisitos e Restrições arquiteturais

|  |  |
| --- | --- |
| **Requisito** | **Solução** |
| Linguagem | O software foi desenvolvido utilizando a linguagem de programação JavaScript com o interpretador Node.js sendo empregado no framework Sails.js. |
| Plataforma | O próprio interpretador Node.js fornece um servidor de aplicação. |
| Segurança | Com o intuito de manter a segurança e integridade dos dados, o acessso ao software é realizado através de um login, onde é verificado a autenticidade do usuário. |
| Persistência | Utilização de um DAO feito em Node.js que armazena os dados em um banco NoSQL MongoDB. |

# Qualidade

A arquitetura de software suporta os requisitos de qualidade, conforme estabelecido:

1. Confiabilidade

A confiabilidade consiste em um requisito de qualidade, onde é definida pela capacidade que um software possui de funcionar sem apresentar falhas dentro de um determinado período de tempo e inserido em um ambiente específico, a falha pode ser entendida como um acontecido proveniente da não conformidade de uma funcionalidade do software, por tanto, o processo de produção do software SysEstoque possui uma etapa para aumentar a confiabilidade, tal etapa consiste no teste da parte a ser entregue no final da sprint.

1. Usabilidade

A usabilidade é o termo usado para um sistema cujo possua fácil entendimento por parte do usuário, de fácil aprendizado, atrativo, portanto, a interface do software SysEstoque é desenvolvida utilizando convenções de interface, contendo utilização de ícones representativos com a vida real para possibilitar o fácil aprendizado, além de um design atrativo para o usuário.

1. Segurança

A segurança em software consiste na capacidade que um software possui de assegurar que, as informações não seram acessadas por usuários e/ou outros sistemas que não estejam autorizados, portanto, o acesso as informações utilizadas pelo software SysEstoque é através de autenticação.

# Visão lógica

O usuário interage com o sistema através da view, realizando alguma operação no sistema, a view solicita a controller a ação desejada pelo o usuário, a controller processa as informações por meio da comunicação com a model e o banco de dados, que por sua vez se comunica com o banco de dados, que, por conseguinte repassa o resultado da operação solicitada para a view.

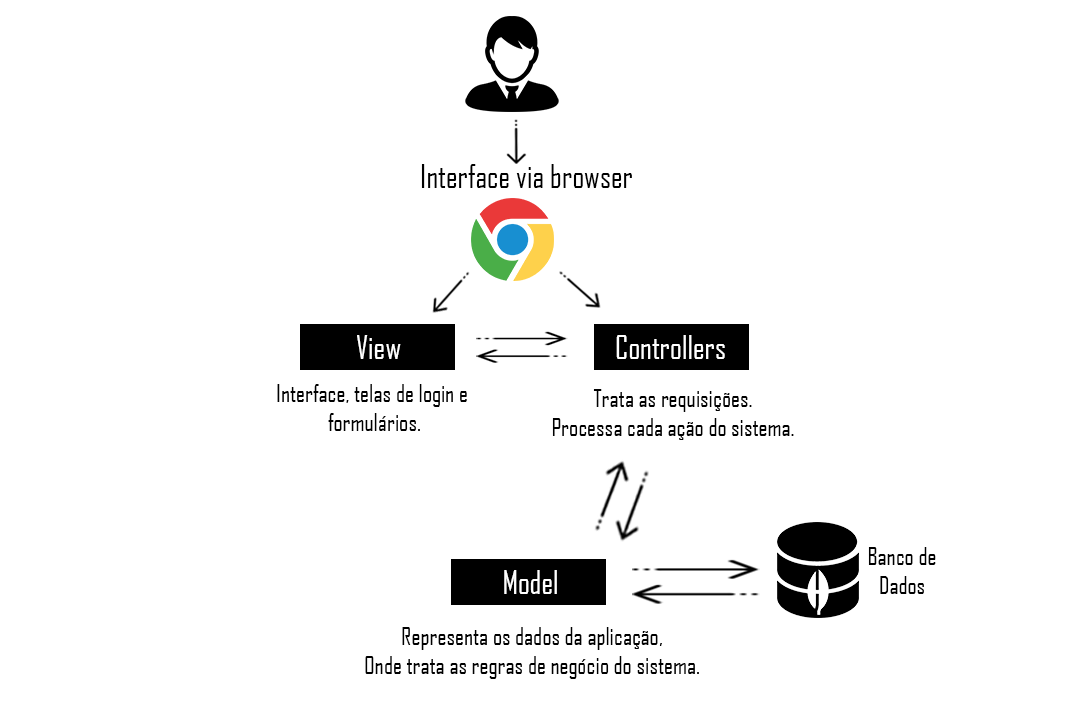


Figura - Visão geral da arquitetura

O sistema SysEstoque é uma aplicação web construída de acordo com o modelo de arquitetura MVC (model-view-controller).

A View é responsável pelo front-end, onde é estabelecida a comunicação entre o usuário e a aplicação. A comunicação acontece apenas com os controllers, através de requisições.

A Controller trata da parte que processa cada ação do sistema, como por exemplo, requisições do usuário. Pode ser entendida como uma ponte de ligação entre a Model e a View, ou seja, a Controller interpreta os eventos que acontecem na camada View, e opera os dados que estão na Model validando os mesmos, estabelecendo a comunicação com o banco de dados a qual fica com a responsabilidade de armazenar estes dados da aplicação.

A camada de Model representa os dados da aplicação, basicamente na camada Model ocorre o tratamento da escrita, validação e leitura dos dados. Estes dados devem estar descritos pelas regras de negócio do sistema. Assim a Model, armazena se necessário os dados no banco de dados, e se comunica com a Controller quando houver necessidade de exibição, e a Controller decidira em qual view exibir os dados da Model.

Na visão lógica os desenvolvedores poderão também compreender a decomposição do sistema em classes, permitindo agregar qualidade ao sistema, além de possibilitar flexibilidade e a extensibilidade do sistema.

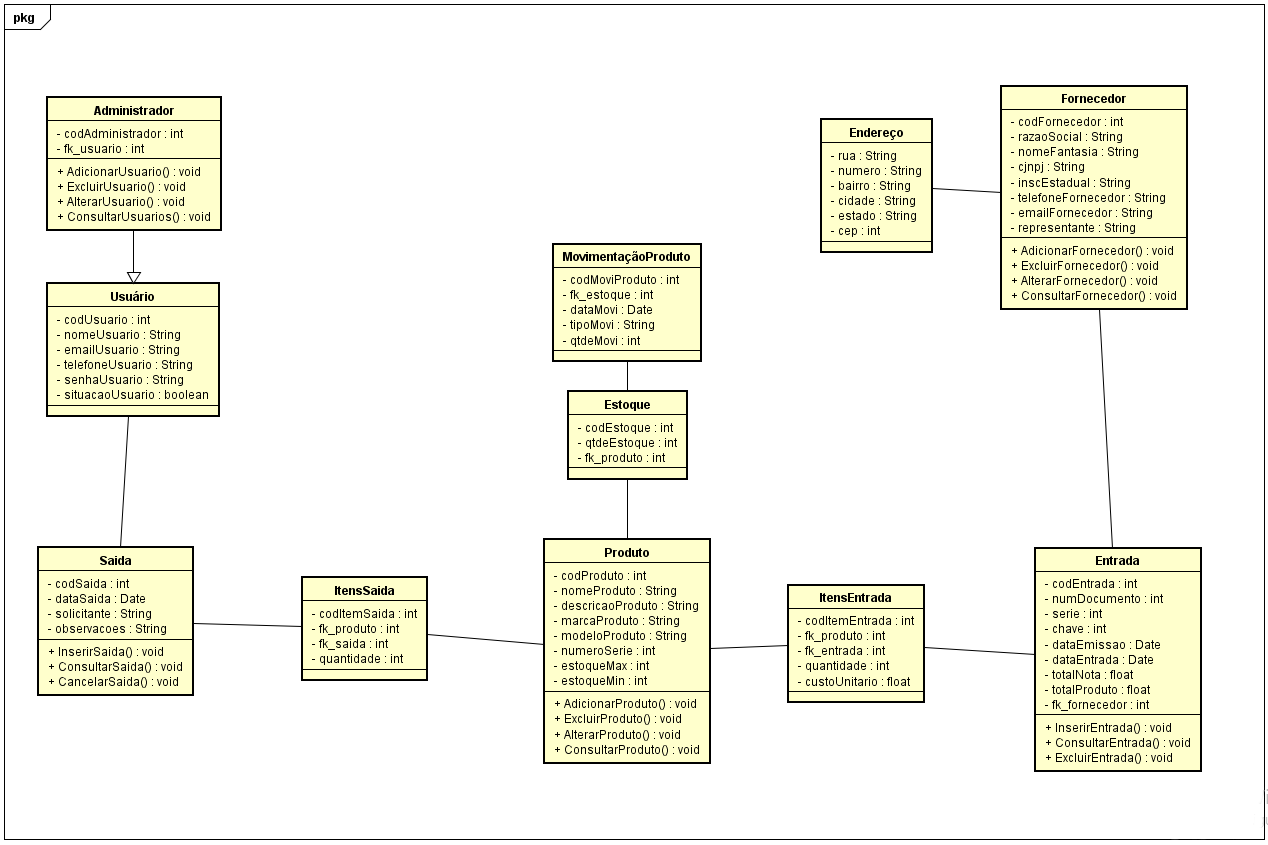


Figura 2- Diagrama de Classe

A visão lógica da aplicação é composta por três elementos principais: Node.js, Sails.js e MongoDB respectivamente o interpretador da linguagem de programação JavaScript, o framework utilizado para o desenvolvimento e o bando de dados utilizado para armazenar os dados.

A API faz requisições através de rotinas e controllers. De forma geral ela é responsável por automatizar as modificações e acessos necessários aos dados. O controller se encarrega de tratar essas requisições e trazer a resposta do banco de dados MongoDB para a API onde a API repassa as informações requeridas para a Viewer.

Devido a utilização da linguagem de programação JavaScript, foi escolhido o interpretador Node.js da mesma linguagem, utilizado para o lado do servidor (server-side). Assim, utilizando o framework Sails.js baseado no padrão MVC para Node.js.

A estrutura básica do front-end, foi desenvolvida utilizando HTML 5. O Bootstrap e JQuery, foram utilizados para desenvolver o layout e o design dos componentes, além de realizar as devidas validações de campos.

# Visão de implantação

Para a implantação do sistema SysEstoque a sua hospedagem será realizada localmente, ou seja, no servidor do próprio setor que pode ser facilmente configurado.

