**Courier**

**Courier**

**Documento de Arquitetura de Software**

**Versão <1.0>**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 05/06/17 | 1.0 | Elaboração do documento | André Costa Ribeiro |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Índice Analítico**

[1. Introdução 4](#_Toc6132)

[1.1 Finalidade 4](#_Toc6133)

[1.2 Escopo 4](#_Toc6134)

[1.3 Definições, Acrônimos e Abreviações 4](#_Toc6135)

[1.4 Visão Geral 4](#_Toc6136)

[2. Representação Arquitetural 4](#_Toc6137)

[3. Metas e Restrições da Arquitetura 5](#_Toc6138)

[4. Visão de Casos de Uso 5](#_Toc6139)

[4.1 Realizações de Casos de Uso 6](#_Toc6140)

[5. Visão Lógica 6](#_Toc6141)

[5.1 Visão Geral 6](#_Toc6142)

[5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 8](#_Toc6143)

[6. Visão de Implantação 8](#_Toc6144)

[7. Visão da Implementação 8](#_Toc6145)

[8. Tamanho e Desempenho 8](#_Toc6146)

[9. Qualidade 8](#_Toc6147)

**Documento de Arquitetura de Software**

# Introdução

O sistema Courier tem como finalidade controlar os recursos que entram e saem em um estoque. O estoque genérico que é controlado pela empresa, pode ser manipulado e verificado levando em consideração a quantidade e o estado de todos os seus itens.

## 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 auxilia os envolvidos no projeto a captar aspectos arquiteturais do sistema que são necessários para o desenvolvimento de um sistema que atenda às necessidades dos usuários finais. Além de auxiliar no entendimento do sistema por novos membros da equipe.

## Definições, Acrônimos e Abreviações

 MVC – Padrão de arquitetura de software onde M significa modelo sendo responsável pela parte de regras de negócio, V a visualização responsável pela parte de interfaces e C a parte de controle dos dados.

## Visão Geral

São apresentados ainda neste documento diferentes visões arquiteturais de como o sistema deve se comportar em diferentes processos, como deve ser implantado é implementado e restrições de desempenho e qualidade.

# Representação Arquitetural

* Visão de caso de uso o Apresenta as funcionalidades arquiteturais importantes e os usuários do sistema.
* Visão lógica o Descreve as classes e sua organização e apresenta o padrão de arquitetura que deverá ser utilizado para o desenvolvimento do sistema.
* Visão de processos o Mostra o padrão de comportamento do sistema diante de diferentes ações do usuário.
* Visão de implantação o Descreve a estrutura do ambiente onde o software será instalado.
* Visão de implementação o Ilustra a distribuição do processamento em um conjunto de nós do sistema, incluindo a distribuição física dos processos e threads.

# Metas e Restrições da Arquitetura

Existem algumas restrições de requisito e de sistema principais que têm uma relação significativa com a arquitetura, sendo elas:

* Utilização do paradigma Orientado a Objetos para o desenvolvimento;
* Estrutura MVC;
* Linguagem de programação PHP;
* O sistema em questão deverá ser Web;
* Banco de dados MySQL;

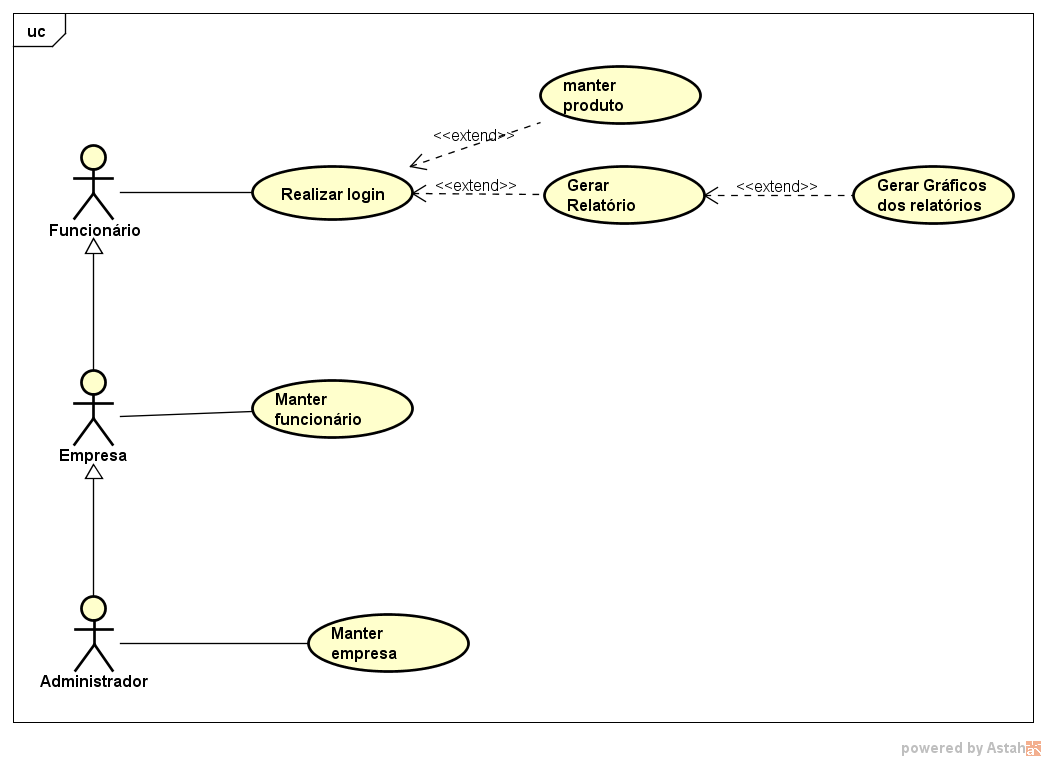
# Visão de Casos de Uso

Os casos de uso do sistema Courier serão listados abaixo:

* Manter estoque;
* Manter Funcionários;
* Manter produtos;
* Gerar relatórios;
* Gerar gráficos dos relatórios;
* Manter empresa;
* Efetuar login;

## Realizações de Casos de Uso

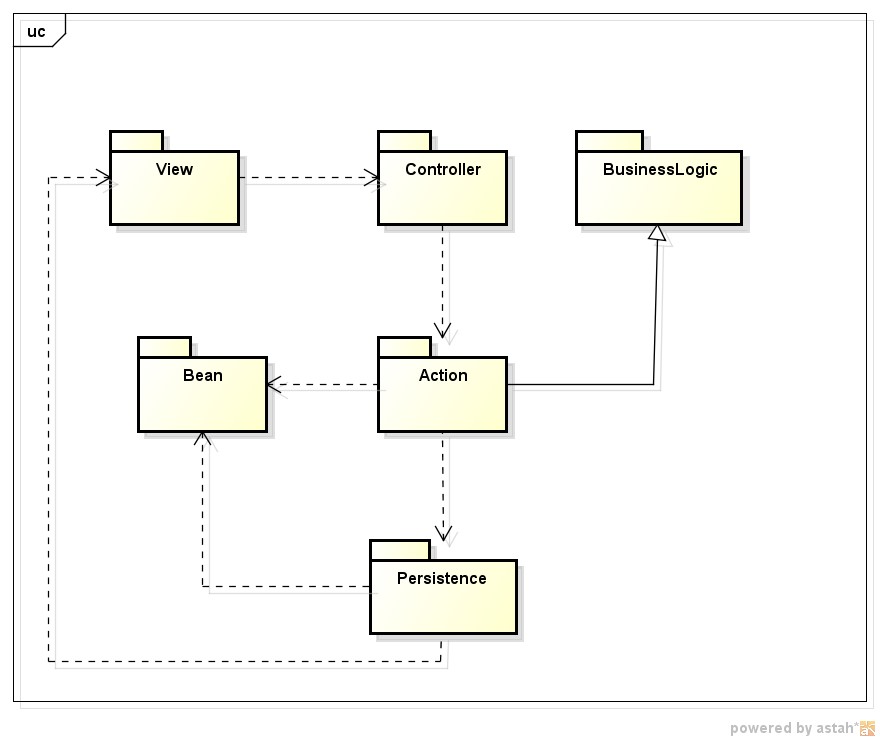
Diagrama de Caso de Uso



# Visão Lógica

## Visão Geral

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso, MVC.

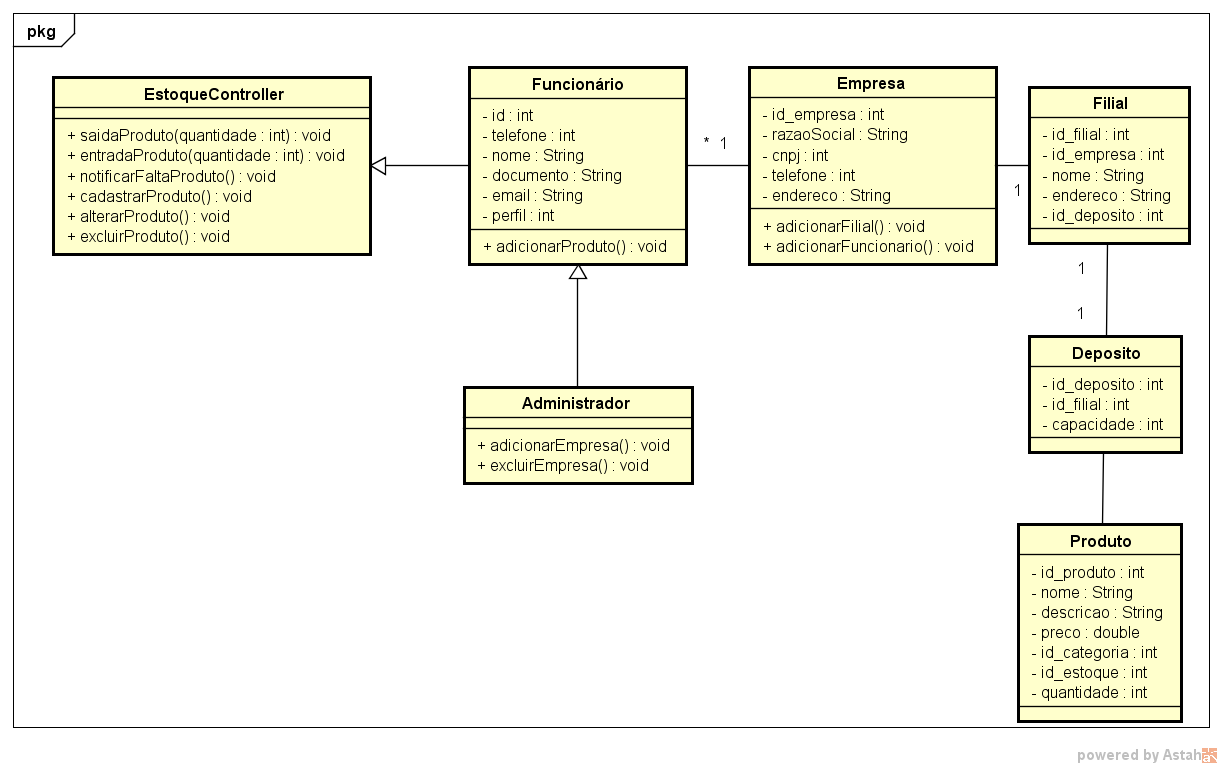


Onde:

* *View*: pacote que contém as visões do projeto, ou seja, as interfaces, formulários e etc.;
* *Controller*: pacote que recebe as informações e requisições do pacote *view* e os despacha para devida classe de controle, o pacote *controller* implementa o pacote *BusinessLogic* o qual contém as regras de negócio do sistema;
* *Action*: pacote que recebe as informações e requisições do pacote *controller* e os atribui às respectivas classes do pacote *bean* e do pacote *persistence*;
* *Bean*: pacote que recebe e armazena as informações referentes a seus respectivos objetos;
* *Persistence*: pacote que recebe as requisições e realiza as operações relacionadas ao Banco de Dados, utilizando dados provindos do pacote *bean*. O pacote *persistence* também pode enviar dados para o pacote *view*;

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

Diagrama de Classe geral do sistema.



# Visão de Implantação

O sistema será implantado no servidor da Fábrica de Tecnologias Turing, no qual será disponibilizado para acesso através da web.

# Visão da Implementação

O sistema será implementado utilizando conceitos de Programação Orientada a Objetos, Estrutura MVC, JSP, *Servlet*, *Web Container Apache Tomcat* e Linguagem de Programação PHP e Banco de Dados MySQL.

# Tamanho e Desempenho

Não há restrições explicitas relacionadas a tamanho ou desempenho.

# Qualidade

O padrão de arquitetura adotado no projeto tem como finalidade garantir uma melhor organização do código-fonte, o que auxilia na manutenibilidade do software, bem como a portabilidade do mesmo.