**Documento de Arquitetura de Software**



**SPD em Plantações**

**Histórico de Versões**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** | **Revisor** |
| 31/05/2017 | 1.0 | Modelagem e desenvolvimento | Gustavo | - |

|  |  |
| --- | --- |
| **Cliente** | FATEC - Interno |
| **Documento** | Documento de Arquitetura de Software: *SPD em Plantações* |
| **Data** | 31 de maio de 2017 |
| **Autor** | **Gustavo Sampaio**  sampaio.gustavo18@gmail.com |
|  |  |

**Página de Assinaturas**

|  |  |  |
| --- | --- | --- |
| Revisado e Aprovado por: |  |  |
|  |  | 31/05/2017 |

**Índice**

Camada de Apresentação 7

Camada de Persistência 11

Diagrama de caso de uso geral do sistema 5

Escopo 4

Objetivo 4

Pacote Model 10

Qualidade 13

Realização dos Casos de Uso Significativos 11

Representação Arquitetural 4

Tamanho e Performance 13

Visão de Dados 13

Visão de Implementação 13

**Objetivo**

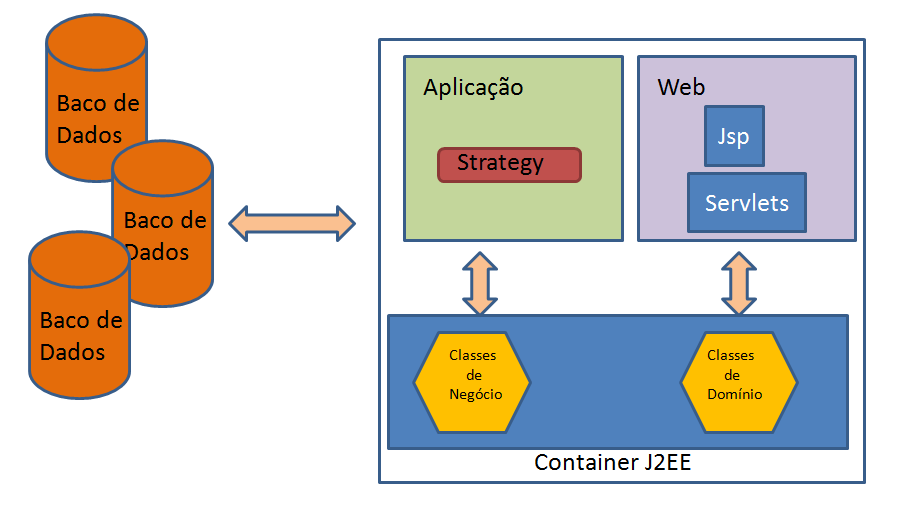
Este documento tem por objetivo apresentar uma visão arquitetural do sistema. O intuito é salientar diferentes aspectos deste produto, obtidos a partir de decisões arquiteturais realizadas no âmbito dos sistemas.

**Escopo**

O escopo deste documento é documentar as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Além disso, mostra sua divisão em classes e utilitários de classe.

**Representação Arquitetural**

Os sistemas serão desenvolvidos tendo como base a arquitetura ilustrada na Figura 1. Toda a arquitetura será baseada nos padrões de projetos tradicionais do GoF e também nos padrões J2EE sendo executados dentro de um Servidor de Aplicações.



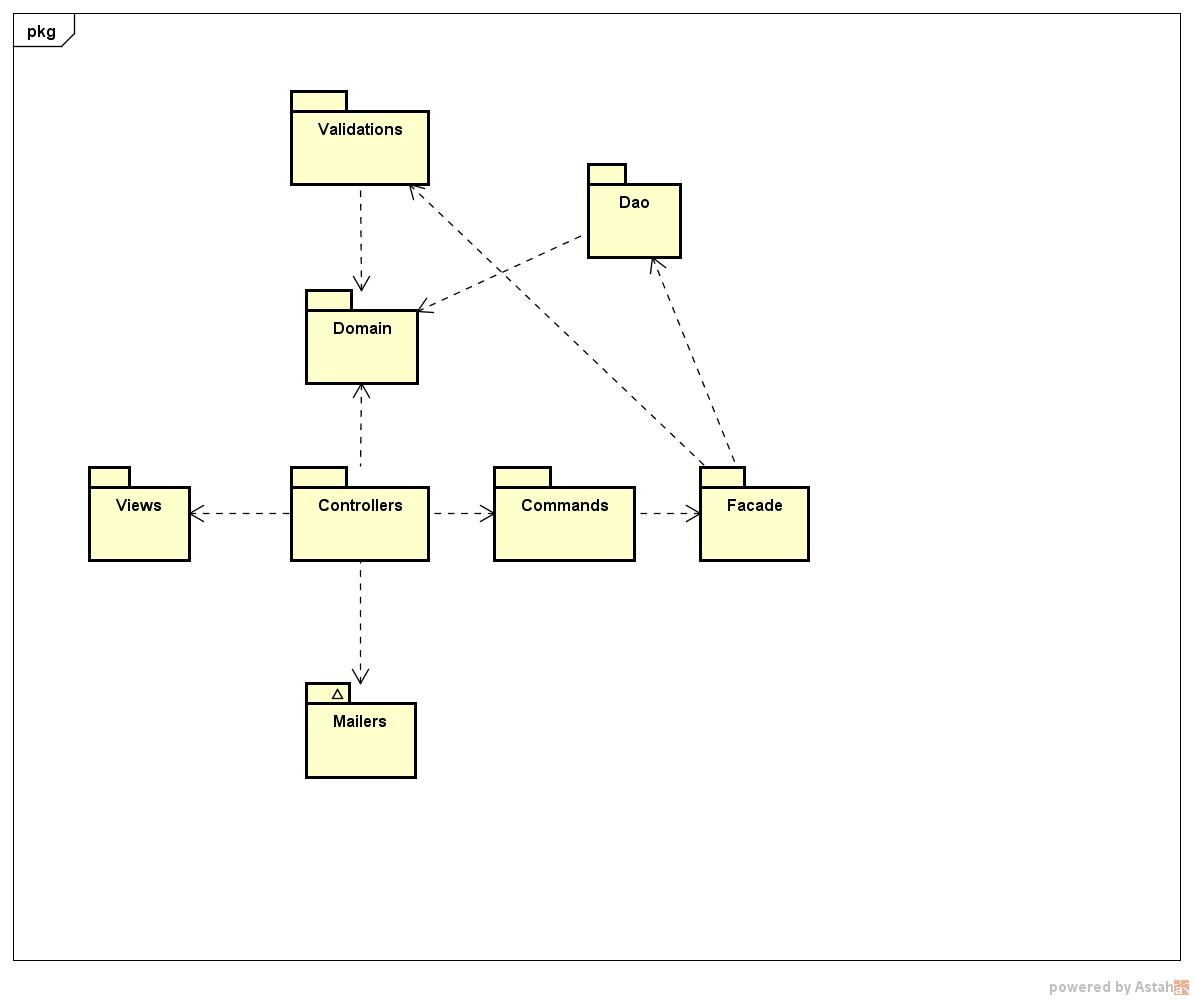
**Figura 1 - Modelo Arquitetural Genérico**

O container “web” conterá os componentes da arquitetura responsáveis pela camada de apresentação. Neste projeto, serão usadas páginas em JSP, com estilos css e scripts em JavaScript, assim como um servlet para interpretar as requisições. Este container também conterá comands e view helpers, que compõem a camada de controle.

As Classes de Domínio são as classes que representam os Value Object, contendo somente os atributos e os métodos getters/setters.

As Classes de Negócio representam as classes responsáveis por aplicar as regras de negócio do sistema como, por exemplo, cadastrar funcionário. Constará também nas classes de negócio o relacionamento com os DAOs responsáveis por persistir e recuperar os objetos no banco de dados.

O container de aplicação conterá as regras de negócio, implementadas usando o padrão strategy. Uma fachada comporá o ponto de contato com a camada de persistência.



**Figura 2 - Diagrama em pacotes**

* 1. Restrições Arquiteturais

Foram identificadas algumas orientações / restrições pertinentes ao desenvolvimento deste subsistema:

* Utilização do JDK 6 do Java;
* Utilização do servidor Oracle Application Server ou TomCat 7.X;
* Utilização do SGBD PostgreSQL.
  1. Objetivos e Restrições Arquiteturais

Alguns requisitos registrados que impactam diretamente a arquitetura do sistema SPD são:

* A consideração da utilização da API de previsões climáticas.

Considerando premissas definidas para os Sistemas SPD pode-se citar as seguintes restrições:

* Utilização da Linguagem Java
* Considerar a utilização de software Livre, quando possível
* Necessidade de comunicação com sistemas API do OpenWeather
* O Sistema de Gerenciamento de Banco de Dados a ser considerado em implementações de âmbito corporativo será o SGBD Postgres.

1. Visão de Use Case

Esta seção apresenta os Casos de Uso arquiteturalmente significativos, que foram selecionados considerando-se o pacote do Modelo de Casos de Uso que representa o sistema SPD.

A classificação dos casos de uso, em termos de significância, foi realizada com base na observação de pelo menos um dos seguintes critérios:

* Casos de uso que estendem outros Casos de Uso
* Casos de Uso que são incluídos em outros Casos de Uso e
* Casos de uso que acessam sistemas externos
  1. Diagrama de Caso de Uso de Condução

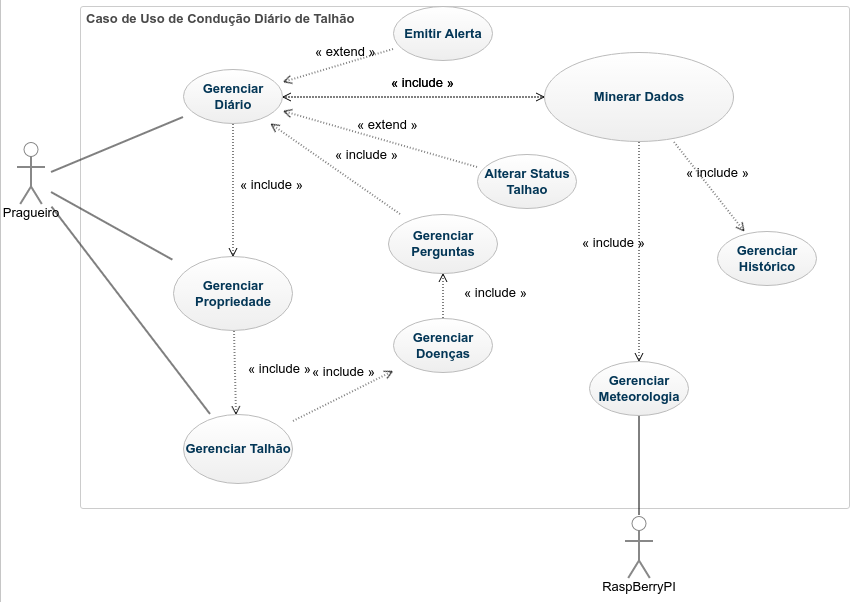
****

Figura 4.1 Diagrama de Caso de Uso de Condução

* 1. Descrição dos Casos de Uso Arquiteturalmente Significativos

Este caso de uso se inicia ao momento em que o usuário escolher qual propriedade e por conseguinte, qual talhão serão coletados as informações. Nesta situação as perguntas relacionadas as doenças que consequentemente estarão atreladas ao talhão serão feitas, onde desta forma ao envio dos dados para o servidor, o sistema irá minerar as respostas e retornar uma porcentagem de chance da doença acontecer na região, dependendo do nível retornado, o sistema emitirá uma mensagem de aviso e alterará o status do talhão.

**Gerenciar Diário**

Este caso de uso representa as funcionalidades do sistema que permitem o administrador consultar e cadastrar um diário

**Gerenciar Propriedade**

Neste âmbito da aplicação, este caso de uso tem como responsabilidade a consulta de propriedades, mas como um todo, o mesmo tem responsabilidade de consultar, alterar, excluir e inserir.

**Gerenciar Talhão**

Da mesma forma do que a propriedade, este caso de uso utiliza apenas a função de consulta neste âmbito.

**Gerenciar Doenças/Perguntas**

Vem com o intuito de carregar as perguntas das doenças sugeridas de acordo com o talhão carregado.

**Minerar Dados**

A partir das respostas das doenças, utilizando o caso de uso de **Gerenciar Doenças** o sistema de encarregará de consultar o arquivo que, posteriormente havia sido feito o upload do mesmo no servidor, o sistema irá utilizar da plataforma WEKA de mineração de dados para que consigo retornar uma reposta lógica sobre a porcentagem de chance da determinada doença acontecer.

1. Visão de Lógica

Esta visão apresenta elementos de design significativos do ponto de vista da arquitetura, descrevendo a organização do Sistema SPD em pacotes, bem como a organização desses pacotes em camadas.

O Diagrama com as camadas do sistema SPD é ilustrado na figura 5.1.



Figura 5.1 – Diagrama de camadas do XXXX

**Apresentação**: Contém classes para as interfaces gráficas com os usuários (GUI). Através destas interfaces os usuários conseguem interagir com o XXXX, com o intuito de incluir, alterar e excluir produtos.

**Negócio**: Contém classes que controlam a execução das funcionalidades do XXXX.

**Persistência**: Contém classes responsáveis por persistir as entidades de modelo. Por exemplo,

contém as classes que permitem ler e gravar os objetos no banco de dados relacional.

Neste momento, é importante ressaltar que a camada de apresentação envolve componentes que são executados na porção servidora e na porção cliente. Na porção servidora, são executados os componentes que montam as páginas html/jsp de resposta e controlam o fluxo de interação com o usuário. Assim, envolve interpretação de páginas jsp, utilização de helpers para montagem das páginas de resposta, delegação de solicitação para a camada de negócio, obtenção de respostas da camada de negócio, dentre outros. Já na porção cliente, estão os códigos javascript que são executados nos navegadores dos usuários com o intuito de facilitar a interação. Por exemplo, podem fazer uma pré-verificação dos dados digitados (como o preenchimento de campos obrigatórios), solicitação de páginas via ajax, de modo a tornar mais dinâmica a interação com o usuário, dentre outros.

* 1. Camada de Apresentação

Nesta camada, temos o pacote form que contém todos os arquivos relacionados à exibição de informações para usuário, o que engloba páginas JSP e HTML, imagens, javascript, dentre outros. Já o pacote de controle desta camada, contém as Actions do Struts responsáveis pela comunicação com as classes da camada de negócio. Estes pacotes podem ser vistos na Figura 5.3.

Figura 5.3: Camada de Apresentação

* + 1. Camada de Negócio

Nesta camada, temos o pacote controle que contém as classes responsáveis por controlar as regras de negócio da aplicação. O pacote model, contém as classes que representam o modelo, ou seja, aquelas que contém as informações sobre o Sistema XXX. O pacote controller contém as classes de controle do negócio. O pacote complementador contém classes que auxiliam na complementação de informações e o pacote filtro contém as classes que auxiliam na filtragem de informações.

A imagem 4.4 ilustra os pacotes descritos.

Figura 5.4: Camada de Negócios

* + 1. Pacote Controller

A figura 5.5 ilustra as principais classes de controle.

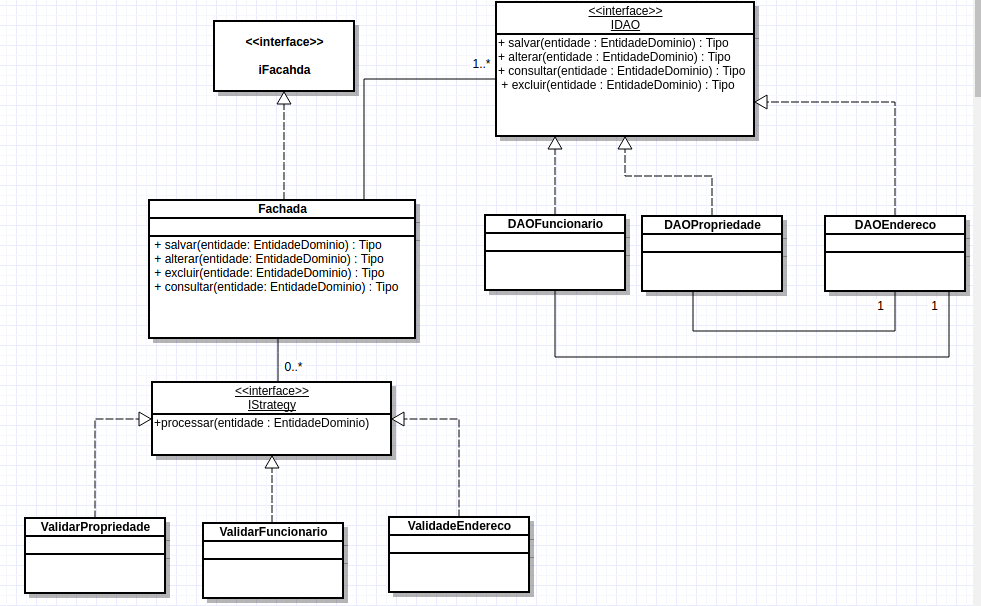


Figura 5.5: Classes de controle

* + 1. Pacote Model

A figura 5.6. ilustra as principais classes do modelo.

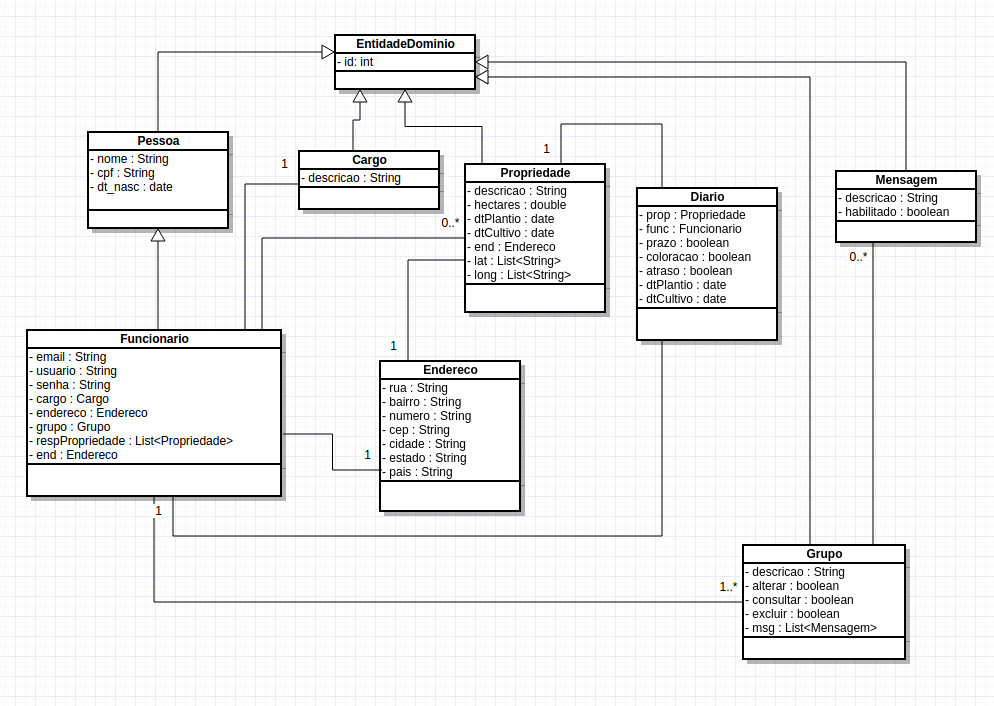
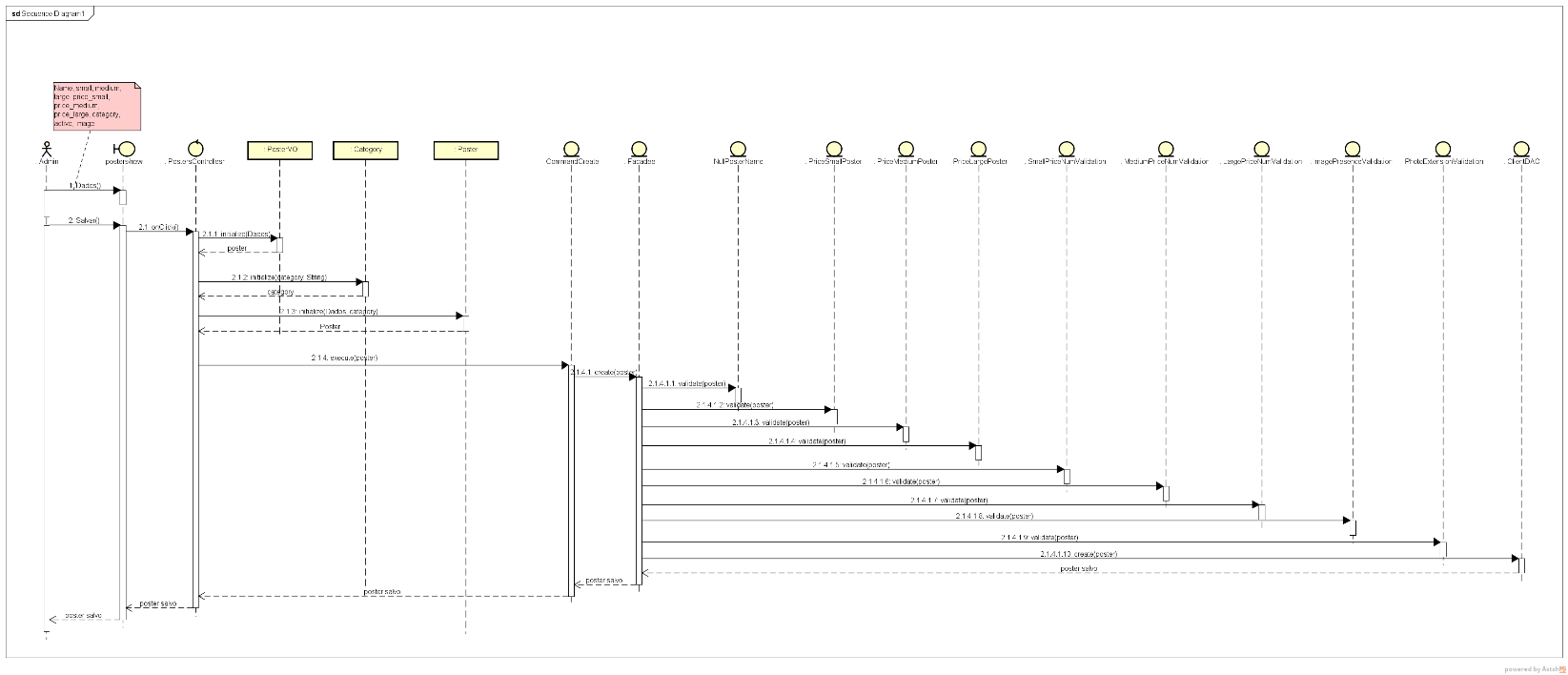


Figura 5.6: Classes do Modelo

* 1. Realização dos Casos de Uso Significativos



1. Visão de Implantação

Esta seção descreve as configurações da rede física (hardware) na qual o XXX será implantado e executado.

Trata-se de uma visão do Modelo de Implantação que, para a configuração em questão, indica os nós físicos (computadores, CPUs), que executarão o subsistema XXXX, e as respectivas interconexões (barramento, LAN, etc). A figura 6 ilustra o modelo de implantação para o XXXX.

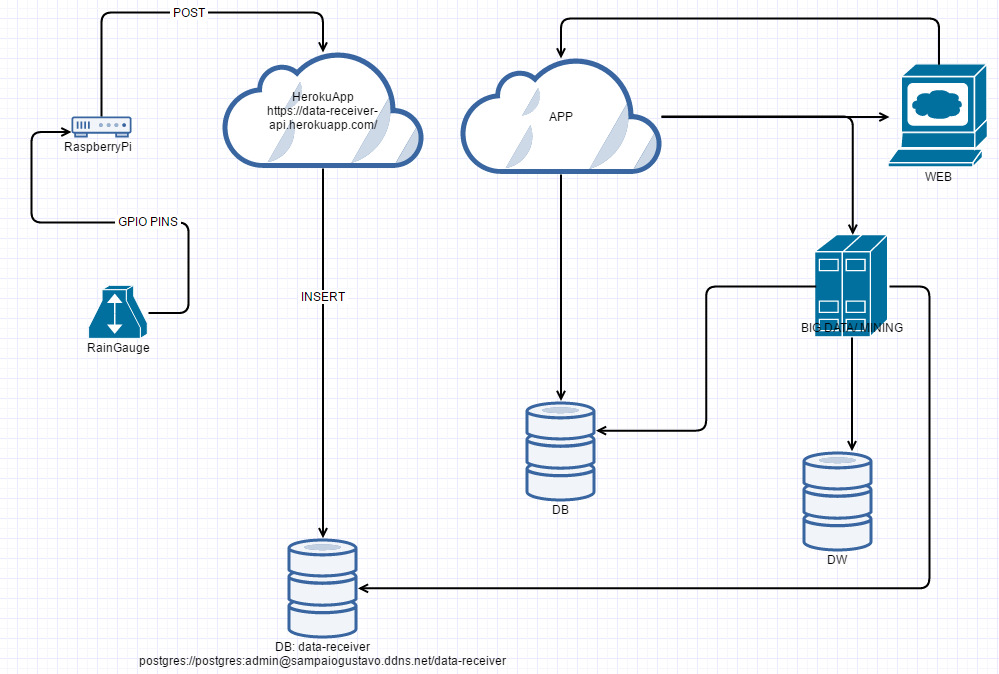


Figura 6: Visão de Implantação do XXXX

Na Figura 6 observa-se os seguintes nós físicos:

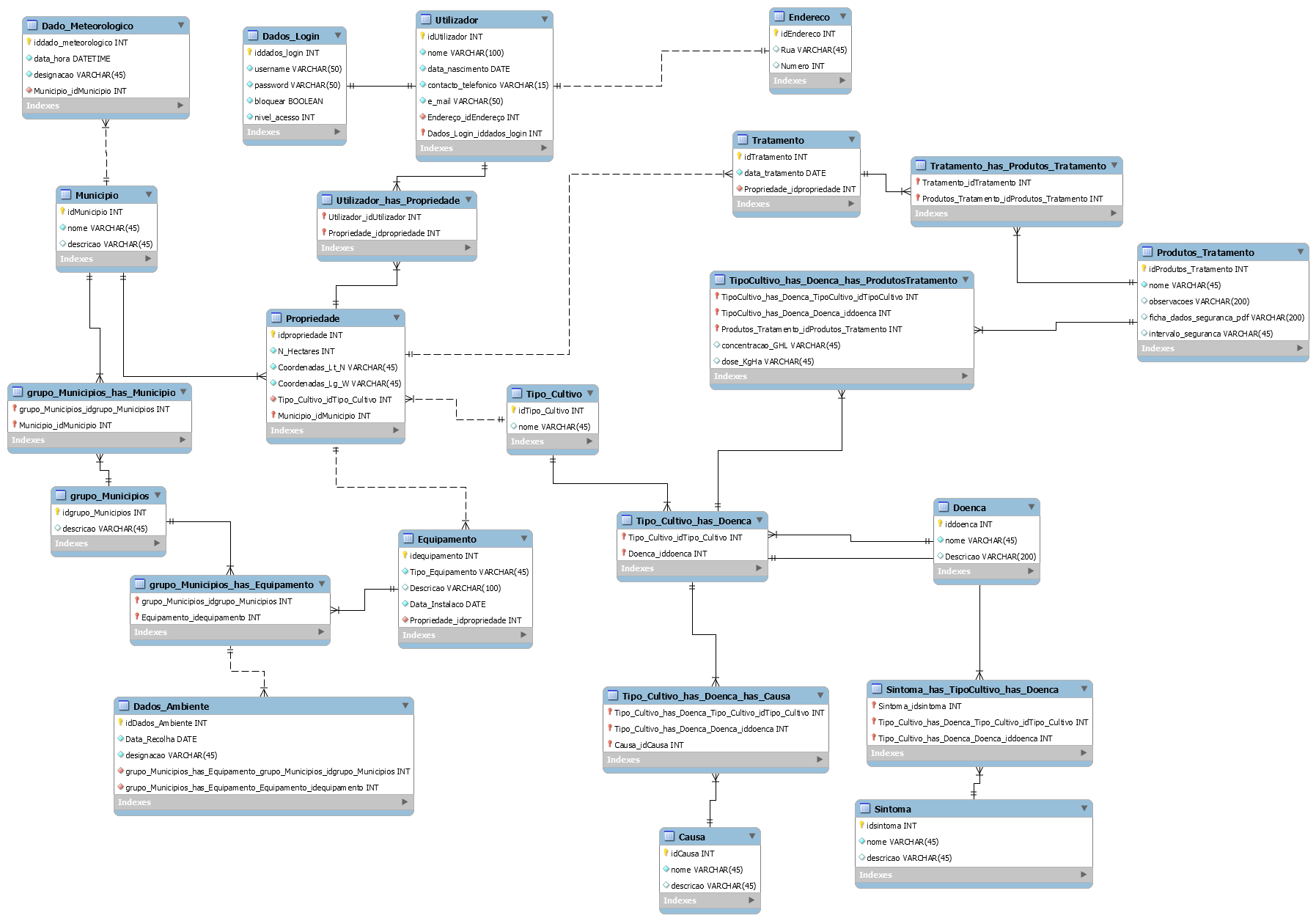
* **RainGauge:** Pluviômetro eletrônico para coleta e envio dos dados para o RaspberryPi
* **RaspBerryPi:** Ferramenta responsável por coletar os dados de dispositivos eletrônicos através dos pinos GPIO's
* **HerokuApp:** API dedicada a recebimento de requisições JSON para, posteriormente efetuar a ação de salvar no banco de dados de meteorologia
* **DB data-receiver:** Banco dedicado para armazenamento dos dados meteorológicos
* **DB:** Banco responsável para armazenamento dos dados da aplicação.
* **APP:** Aplicação web desenvolvimento sob a arquitetura citada no documentos

1. Visão de Implementação

Esta visão descreve a estrutura geral de implementação, a decomposição do software em camadas de implementação.

A estrutura geral de implementação para o SISCAD é baseada na estrutura da Visão Lógica, assim, não há necessidade de detalhar os diagramas de camadas e pacotes de implementação, uma vez que são fortemente baseados naqueles desenvolvidos para Visão Lógica.

1. Visão de Dados



Figrua 8.1 – Modelo Físico

1. Tamanho e Performance

O sistema SPD será usado para auxílio na tomada de decisões ao aplicar fungicidas e defensivos nas plantações.

Como seu uso é interno, os servidores não passarão por problemas de instabilidade, sendo assim simples sua estruturação e mensuração.

1. Qualidade

O sistema SPD será usado para a realização de vendas de pôsteres online.

Eventuais erros e/ou falhas na sua operação podem levar a prejuízos financeiros significativos, portanto na fase de design deve-se levar em consideração como fatores prioritários a confiabilidade e robustez do sistema.

Adicionalmente, o sistema SPD pode ser alvo de ataques de “hackers” para roubar ou simplesmente corromper informações, possibilidade aumentada pela interface do sistema estar disponível na Internet. Para evitar que tais ataques sejam bem-sucedidos uma infra-estrutura de segurança deve ser especificada e projetada.

1. Referências

Unified Modeling Language: <http://www.omg.org/technology/documents/formal/uml.htm>

RUP. Rational Unified Process.