# Asa

Documento de Arquitetura de Software

# 1. Introdução

Este documento apresenta a arquitetura proposta para o Software pessoal de gestão de atividades Asa. A arquitetura é apresentada através de um conjunto de visões, que juntas tem por objetivo cobrir os principais aspectos técnicos relativos ao desenvolvimento e implantação do sistema em questão.

## 1.1 Finalidade

Este documento visa apresentar a arquitetura geral do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do mesmo. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas no decorrer do processo de desenvolvimento.

## 1.2 Escopo

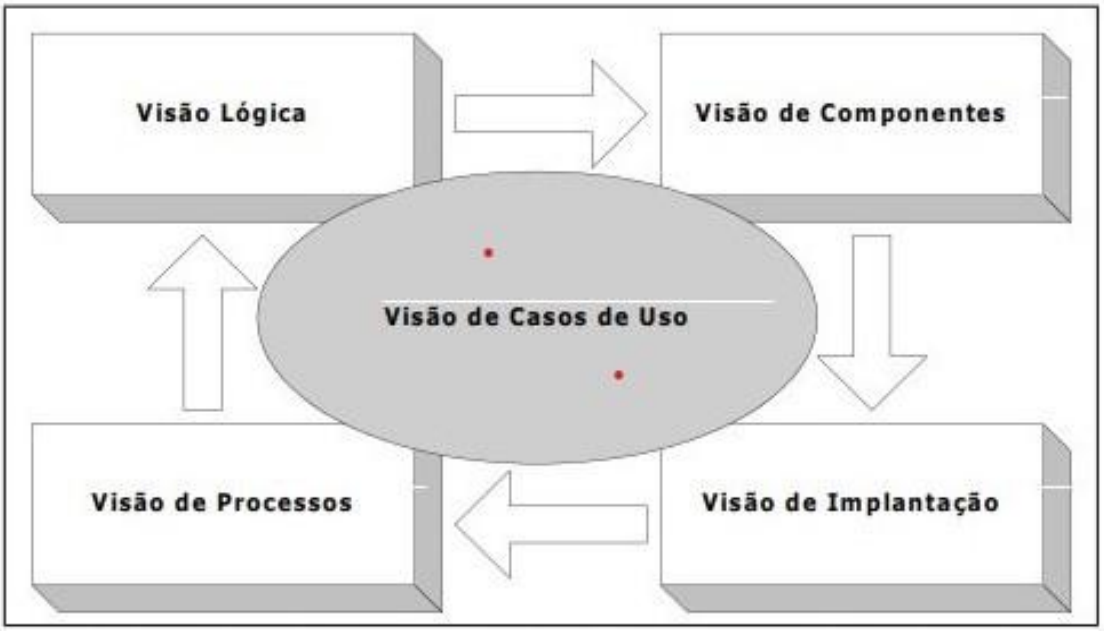
Este documento auxilia os envolvidos no projeto a captar aspectos arquiteturais do sistema que são necessários para o desenvolvimento de uma solução que atenda às necessidades dos usuários finais. Além de auxiliar no entendimento do sistema por novos membros da equipe.

## 1.3 Visão Geral

Outro aspecto que será discutido será a qualidade do sistema garantida pela arquitetura, seja usabilidade, segurança, disponibilidade e manutenibilidade.

# 2. Representação Arquitetural

São apresentados ainda neste documento diferentes visões arquiteturais do sistema de acordo com o Modelo 4+1 de Visualização de Arquitetura (The 4+1 Model of View Architeture ), essas sendo a visão de caso de uso, a lógica, processo, implementação e implantação.




- Visão de caso de uso
  Visão de Casos de Uso O propósito desta visão é apresentar os casos de uso arquiteturalmente significativos para o sistema, ou seja, o conjunto de casos de uso que direcionarão a arquitetura do sistema como um todo. Esta visão usará diagramas de casos de uso para expor as funcionalidades e diagramas de sequência para mostrar a interação entre objetos, contemplando ainda o relacionamento entre os diversos elementos da arquitetura.

- Visão Lógica
  Esta visão irá apresentar as definições do sistema através de diagramas de classes, ou quaisquer outros diagramas que descrevam os serviços que o sistema fornecerá aos seus usuários. Esta visão será utilizada para apresentar a arquitetura em um nível elevado de abstração.

- Visão de processos
  Esta visão mostra a decomposição do sistema, bem como as formas de comunicação entre processos, passagem de mensagens, atividades entre componentes e sequência de mensagem. Essa representação é feita através dos diagramas de sequência, e Arquitetura de Software 7 © 2015 Empresa Brasileira de Pesquisa Agropecuária opcionalmente pelos diagramas de colaboração e atividades.

- Visão de implantação
  Esta visão descreverá como o sistema será distribuído e implantado nos nós físicos, nos quais será executado.

- Visão de implementação

Esta visão irá descrever a organização dos subsistemas e componentes do sistema. Esta visão irá mostrar as diversas camadas do software, bem como as fronteiras entre essas camadas.

# 3. Metas e Restrições da Arquitetura

A arquitetura deve obedecer as metas e restrições impostas pelos requisitos funcionais e não funcionais. Dessa forma, algumas metas a serem tomados como base para as decisões arquiteturais são:

1) Obter ganhos expressivos de produtividade no desenvolvimento dos sistemas;
2) Garantir a evolução dos produtos utilizados como base da arquitetura;
3) Padronizar o desenvolvimento, para facilitar a manutenção.
4) Maior estímulo à utilização de testes automatizados;

### 3.1 Itens restritivos da arquitetura

Os itens restritivos da arquitetura são:

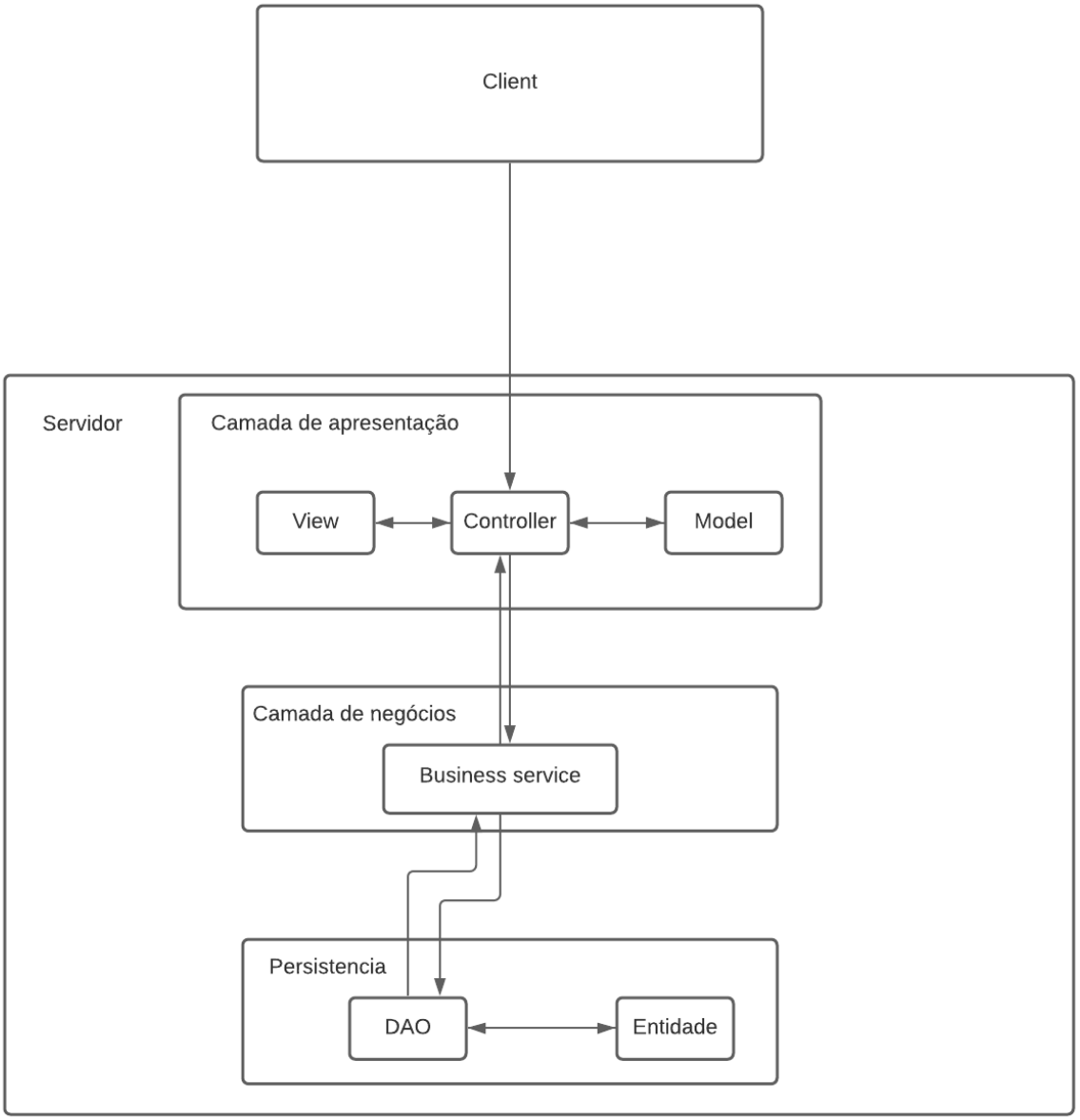
1) Utilização de MySql para banco de dados.
2) Desenvolvimento em estações Linux ou Windows.
3) O sistema desenvolvido será web, em que tanto o frontend quanto o backend serão feitos com js, node.js .

# 4. Visão de casos de Uso

Nesta seção há definição dos principais casos de uso que representam um grande risco arquitetural, no entanto, como a proposta desse documento é de âmbito geral, essas definições serão especificadas em um documento suplementar localizado no mesmo diretório do documento atual.

# 5. Visão Lógica

Esta visão descreve em que arquitetura o software será construído, abordando aspectos de organização dos pacotes, como responsabilidades, topologia e a forma de comunicação e dependência entre os pacotes. Os diagramas abaixo mostram as classes, os pacotes e as camadas mais significativas do ponto de vista arquitetural do framework de arquitetura.




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

1) Client

O client, se refere ao browser na máquina do cliente, aquele que fará o request http ao controller, e esse validando o pedido e retornando uma view.

2) Servidor

O servidor é aquele que recebe os pedidos do client, retornando, se válido, uma view. Será usado javascript, html, css.

2.1) Camada de apresentação

2.1.1) View

Armazena as diferentes telas e eventos do sistema.

2.1.2) Controller

Componente que gerencia as chamadas do servidor, sejam internas ou externas, direcionando a view ou serviço solicitado, ou seja, navegabilidade dos casos de uso e tratamento das exceções oriundas da camada de negócio.

2.1.3) Model

São as entidades do sistema que serão usadas pela view.

2.2) Camada de negócio

2.2.1) Business service

Business service contém as regras de negócio por entidade de negócio, além de fazer as requisições a camada de persistência para salvar informações.

2.3) Camada de Persistência

2.3.1) DAO (Data Access Object)

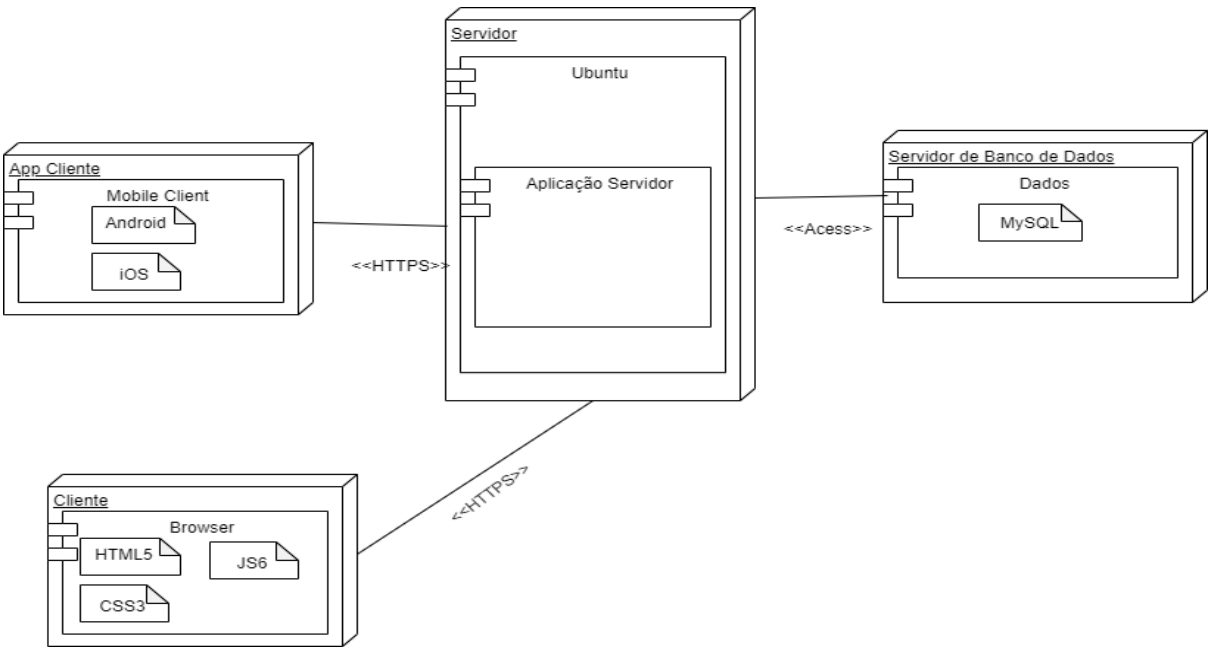
Responsável por realizar a persistência e consulta aos dados.

2.3.2) Entidades

Responsável pela representação dos dados persistentes da aplicação.

# 6. Visão de Implementação

O sistema será implementado utilizando conceitos de Programação Orientada a Objetos através das linguagens de JavaScript, HTML5, CSS3 e Bancos de Dados Firebase e MySql.



# 7. Qualidade

Três principais fatores que influenciam a qualidade de projetos de TI baseados na WEB são: confiabilidade, usabilidade e segurança. Além destes, alguns fatores importantes seriam: disponibilidade e manutenabilidade. A arquitetura apresentada atende a esses requisitos não funcionais da seguinte forma:

1) Confiabilidade

A arquitetura garante a confiança de que a aplicação sempre retornará aquilo que é esperado da mesma. Mantendo a separação das funcionalidades, acesso ao banco, regras de negócio e apresentação, é possível diminuir a ocorrência de erros e a falta de clareza de conceitos

2) Usabilidade

A interface com o usuário será definida da forma mais intuitiva respeitando critérios de acessibilidade possibilitando facilitar o treinamento dos usuários e a gestão de mudança. Como há uma camada especialmente definida para isso, o foco será mantido na solução desta questão separadamente, sem a mistura com outras questões de responsabilidade de outras camadas.

3) Segurança

A utilização de mecanismos de autorização/autenticação unificados em conjunto com uma padronização do sistema de log e auditoria torna possível incrementar a segurança facilitando o desenvolvimento e estabelecendo normas com um baixo esforço dos programadores.