Documento de Especificação de Arquitetura

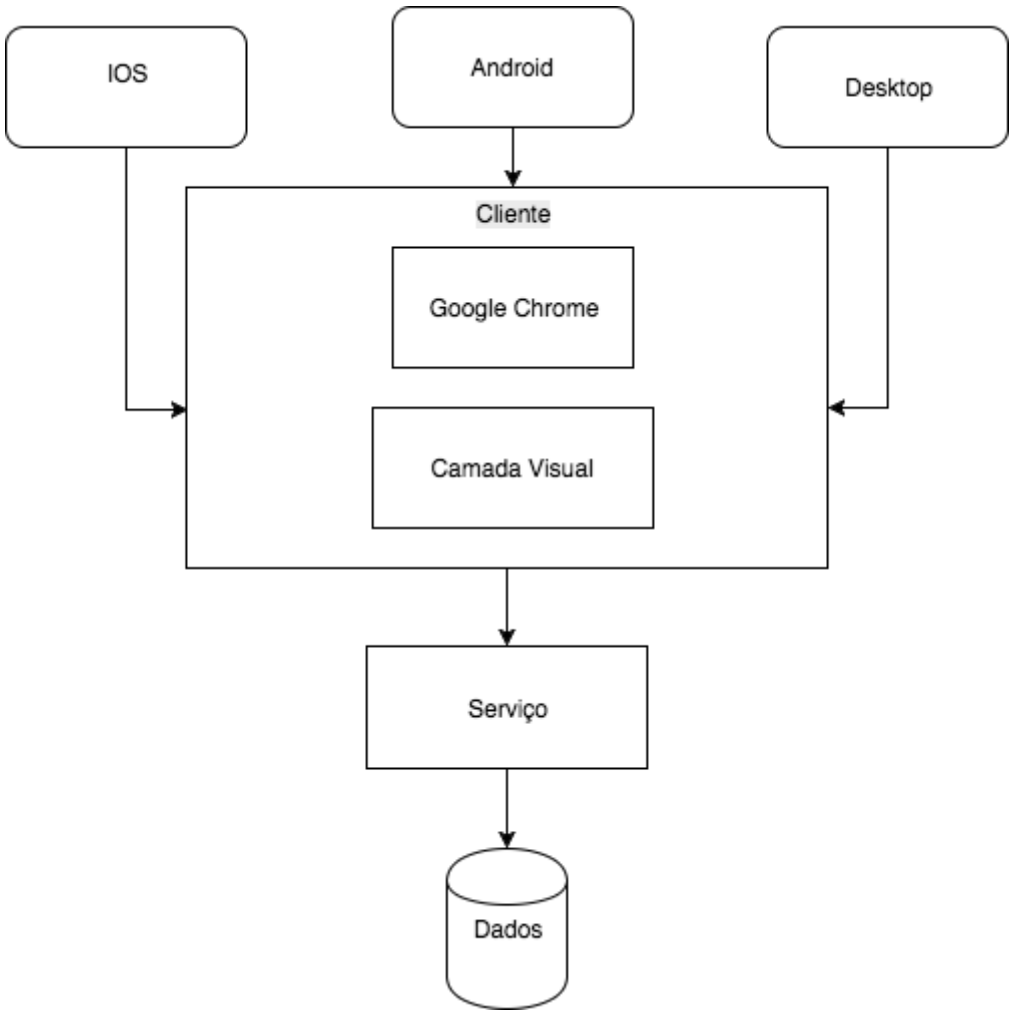
Sistema: Vou the Van

1 Intrudução

Esta documentação tem como objetivo detalhar a arquitetura do projeto Vou the Van, detalhando os requisitos não funcionais, componentes e tecnologias empregadas no escopo do projeto.

2 Visão Geral

O modelo arquitetural que será adotado no projeto é embasado no modelo de cliente servidor, onde a camada do cliente, dependerá do uso dos recursos do navegador Google Chrome, este por vez, irá permitir que o mesmo seja portado para qualquer plataforma que dê suporte a está aplicação.



1. Arquitetura empregada no Software Vou the Van

3 Requisitos não-funcionais

3.1 Desempenho	3.1.1 As páginas serão ***minificadas,*** reduzindo seu conteúdo e tempo de tráfego pela rede, tornando assim as páginas mais leves; 3.1.2 Informações de acesso recorrente serão armazenadas em storages do navegador, diminuindo a requisição de informações ao servidor.
3.2 Portabilidade	3.2.1 Camada visual será desenvolvida de maneira a se adaptar ao tamanho da tela do dispositivo, seja ele tablet, desktop ou smartphone; 3.2.2 A aplicação será multiplataforma, tendo apenas como dependência a instalação do navegador Google Chrome, que por sua vez, se encontra disponível na maioria das plataformas modernas
3.3 Interoperabilidade	3.3.1 O sistema será desenvolvido em Java, sendo esta uma plataforma de desenvolvimento portável entre sistema operacionais que suportam sua máquina virtual, tornando simples a migração de servidores de aplicação e comunicação com outras tecnologias; 3.3.2 A comunicação com o banco de dados será realizada através da API de persistência Java JPA. Ao implementar esta tecnologia, torna-se fácil portar o banco de dados para outros serviços de armazenamento, pois o acesso aos dados será realizado através desta interface heterogênea.

4 Mecanismos Arquiteturais

Mecanismos de Análise	Mecanismo de Design	Mecanismo de Implementação
Persistência	Banco de Dados Relacional	PostgreSQL,
Back-End	Camada Lógica responsável pelo implementação do sistema	Java, utilizando como Frameworks JSF, JPA e CDI.
Camada de persistência	Classes responsáveis por abstrair os modelos relacionais e realizar a persistência e a comunicação com o banco de dados	JPA utilizando Hibernate como provider, implementando o padrão DAO (Data Access Object) com entidades (Entity).
Camada Visual	Camada de cliente responsável por estabelecer uma interface de comunicação entre o sistema e o usuário final	JSF com o framework Primefaces.
Camada de Teste	Responsável pela validação e qualidade no cumprimento do requisitos.	JUnit
Servidor de Aplicação	Ambiente no qual a aplicação será executada e distribuída	Container JBoss em um servidor Amazon Linux

5 Fundamentação

O modelo arquitetural a ser empregado no sistema será o MVC (Model View Controller), com o intuito de diminuir o acoplamento e responsabilidades de cada camada. Foi cogitado o modelo de microservices. Entretanto, por ser tratar de uma aplicação de poucos acessos e com foco em público regional, o modelo MVC seria

o mais viável, tendo em vista que a manutenção em apenas uma aplicação teria menor custo do que manutenir vários serviços interligados.

A linguagem adotada será o Java, por ser uma linguagem de código aberto e com ótimo suporte da comunidade.

Tendo como base a linguagem Java, os frameworks adotados para o desenvolvimento desta aplicação foram: JPA para persistência, utilizando como provider o Hibernate, devido a extensa documentação e maior adoção da comunidade, para camada visual, foi adotado o JSF, devido a ser uma especificação Java para desenvolvimento de aplicações EE, contendo ainda uma bibliotecas de componentes de baixo custo e ótimos benefícios como o **Primefaces**, este também está presente no escopo do projeto.

Como controle de permissão, será implementado o framework Spring Security, lidando com o acesso às requisições e autenticação de usuários.

6 Definição Arquitetural

Os sistema conforme especificado será dividido em 3 (três) camadas, visando diminuir o grau de acoplamento, sendo separadas em **camada de persistência**, **camada de controle** e **camada de visualização**. Cada camada terá apenas acesso a sua camada mais próxima, seguindo a seguinte ordem:

Visualização
Controle
Persistência

6.1 Camada de persistência

Camada responsável por efetuar a comunicação com o banco e mapear as estruturas das tabelas presentes no banco de dados em modelos relacionais de objetos. As operações de inclusão, consulta, atualização e deleção de dados, são realizadas através do módulo DAO (Data Access Object), utilizando para tal o componente JPA.

Os modelos presente nessa camada, são abstrações das estruturas de dados presente no banco de dados, facilitando a manipulação deste através da aplicação e a alteração da estrutura de dados, tornando flexível mudanças futuras.

6.2 Camada de controle

Camada de software com atribuições referente a implementação de operações especificadas no modelo de domínio, integrando a camada de visualização com o camada

de dados do sistema. Nesta camada se encontram as chamadas de operações que visam atender os requisitos funcionais.

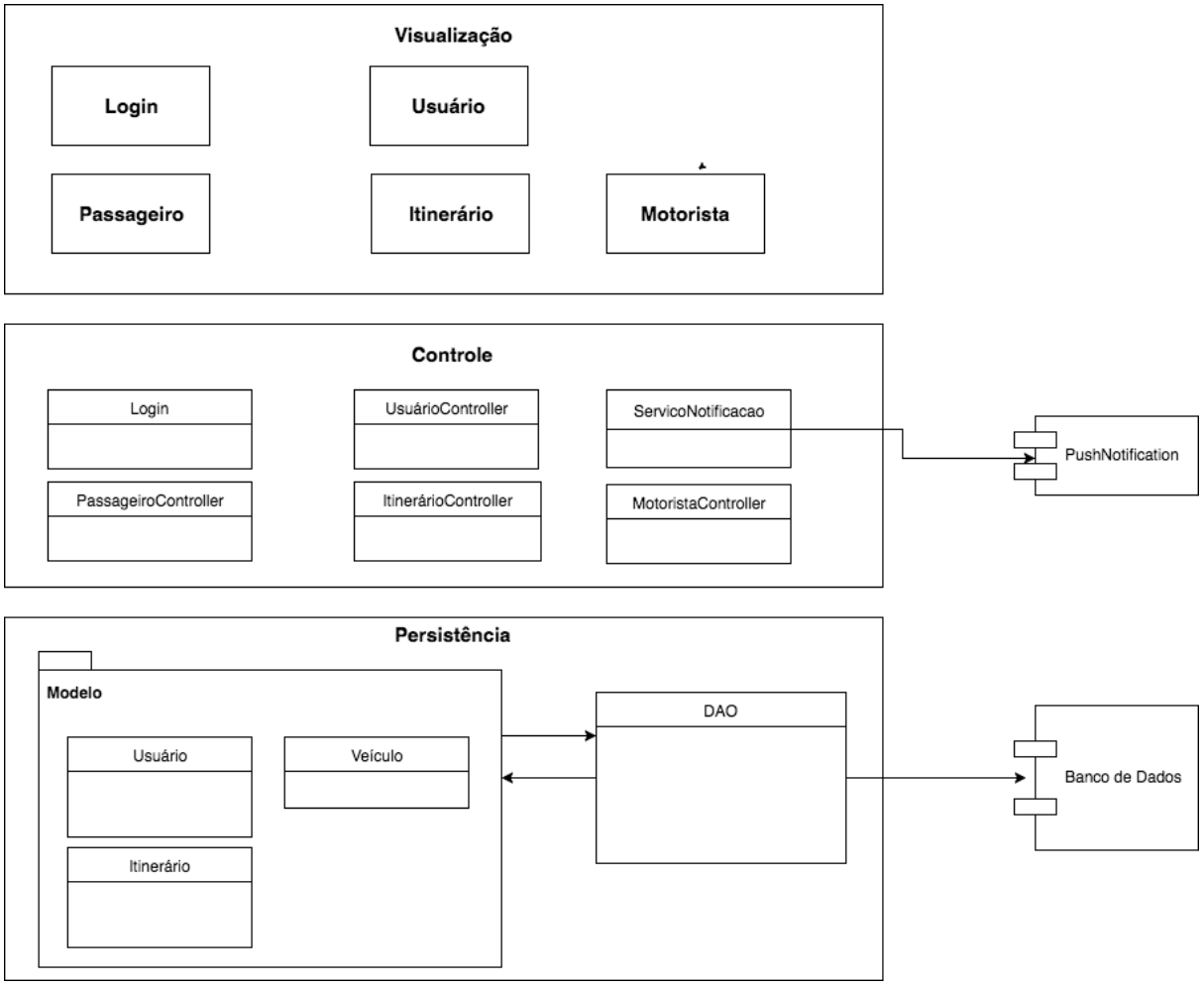
Para cada página do sistema (Camada de Visualização) será implementado um controle, que será responsável por implementar as funcionalidades da respectiva página. Cada controle terá o mesmo nome da sua camada visual, acrescida de um sufixo **Controller**.

6.3 Camada de Visualização

Camada que provê a interação entre o usuário e o sistema, responsável por realizar as chamadas de execução da camada de controle e exibir informações e respostas vindas desta. A camada de controle no sistema Vou The Van será composta pelo conjunto de páginas responsável por compor a interface de comunicação com o usuário final, o passageiro e motorista. Por definição do modelo arquitetural implementado, cada página deve ser controlada por uma classe de controle.

6.4 Diagrama de Módulos

Modelo visual da arquitetura de módulos que deverão compor a primeira release do sistema, podendo esta ser passível de alterações ao longo do ciclo de vida do projeto.



1. Diagrama de módulos