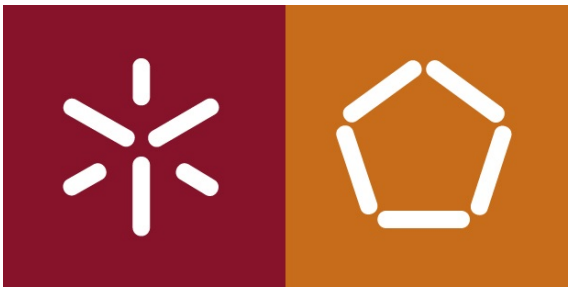

Universidade do Minho

Mestrado em Engenharia Informática

Engenharia Web

Engenharia de Sistemas de Software

Sidewalk Monitoring System - Arquitetura

Francisco Freitas (A81580)
Inês Alves (A81368)
Pedro Freitas (A80975)

2 de Maio de 2020

Resumo

No âmbito da Unidade Curricular de Engenharia Web, enquadrada no perfil de especialização de Engenharia de Sistemas de Software, estamos a desenvolver o projeto de Sidewalk Monitoring System. Neste documento vamos propôr uma solução para a Arquitetura do sistema.

Conteúdo

1 Arquitetura - Panorama geral

Na Figura 1 está presente o panorama geral da arquitetura do nosso sistema. Podemos identificar quatro grupos distintos: *User Interface*, *API*, *Services* e *Database*. Esta é uma arquitetura orientada aos serviços de forma a explorarmos as vantagens que esta nos traz, como por exemplo: escabilidade, disponibilidade, e facilidade de manutenção e testes dos serviços.

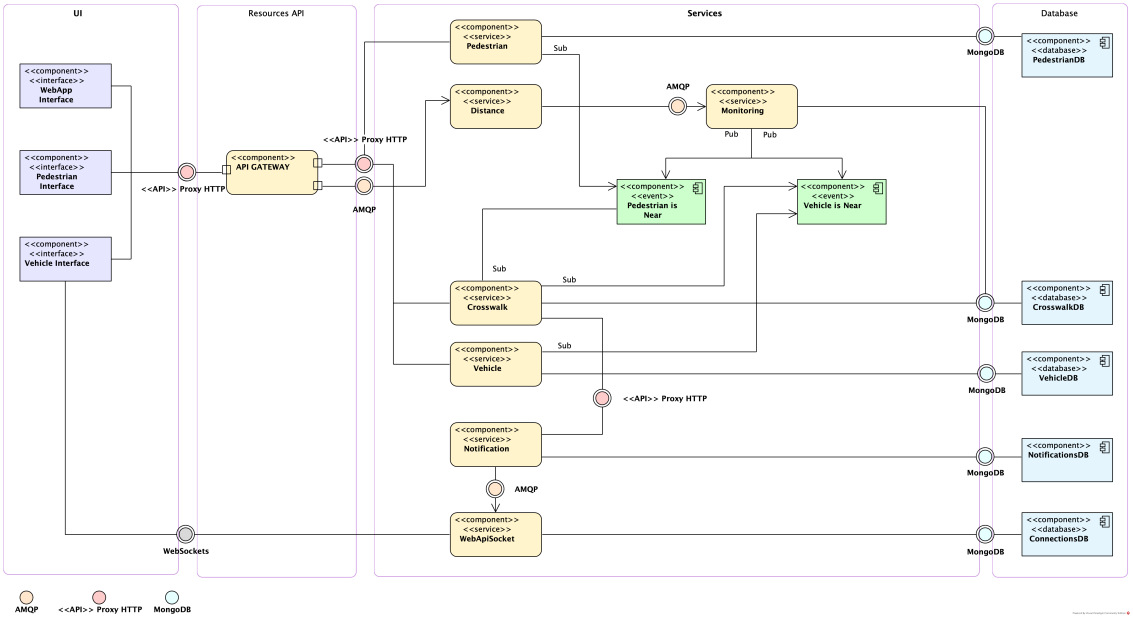

Figura 1: Arquitetura

1.1 Componentes

1.1.1 User Interface

Este grupo diz respeito, tal como o nome indica, à interface da aplicação. É composto por três componentes diferentes:

- ***WebApp*** - Diz respeito à página web geral do sistema
- ***Pedestrian*** - Corresponde à interface da aplicação vista de um peão
- ***Vehicle*** - Corresponde à interface da aplicação vista por um veículo

1.1.2 API

Este grupo é composto apenas por uma componente que funciona como API. Esta é responsável por receber os pedidos provenientes da interface e redirecionar para os serviços apropriados.

1.1.3 Services

Este grupo é responsável pela implementação de todas as regras lógicas que garantem as funcionalidades da aplicação. É composto por:

- ***Crosswalk Service*** - Serviço responsável por gerir as informações relativas a passadeiras.
- ***Pedestrian Service*** - Serviço responsável por gerir as informações de um peão.

- ***Vehicle Service*** - Serviço responsável por gerir as informações de um veículo.
- ***Monitoring Service*** - Serviço responsável por gerir a monitorização dos peões e de carros que estão perto de passadeiras.
- ***Distance Service*** - Serviço responsável por calcular a distancia de um peão ou carro a uma passadeira.
- ***Notification Service*** - Serviço responsável por criar e enviar as notificações aos carros.
- ***WebAPISocket Service*** - Serviço responsável por ser o meio de comunicação entre as notificações e os veículos.

1.1.4 Database

Este grupo é responsável por armazenar todas os dados relevantes para o bom funcionamento do sistema.

1.2 Comunicação

O nosso sistema usufrui dos dois tipos de comunicação, *síncrona* e *assíncrona*, de forma a podermos aproveitar as vantagens que cada uma delas nos traz.

1.2.1 Comunicação Síncrona

De uma comunicação síncrona espera-se que uma entidade envolvente espere por uma resposta sempre que envia uma mensagem a outra entidade. Assim na nossa arquitetura temos comunicação síncrona entre:

- ***CrossWalk Service - API***
- ***Pedestrians Service - API***
- ***Vehicle Service - API***
- ***Crosswalk Service - Notification Service***

1.2.2 Comunicação Assíncrona

Agora com uma comunicação assíncrona uma entidade envia uma mensagem a outra sem esperar uma resposta. Isto permite-nos usufruir de *queues* ou até ativar comportamentos quando um determinado evento ocorre. No nosso sistema veremos comunicação assíncrona entre:

- ***API - Distance Service*** : Via AMQP
- ***Distance Service - Monitoring Service*** : Via AMQP
- ***Monitoring Service - Pedestrian Service*** : Via evento
- ***Monitoring Service - Vehicle Service*** : Via evento
- ***Monitoring Service - Crosswalk Service***
- ***Notification Service - WebAPISocket Service*** : Via AMQP

1.3 Estilos de Arquitetura

Os estilos de arquitetura entre componentes são outro ponto importante para garantir as funcionalidades. Assim temos dois tipos de estilos de arquitetura entre componentes que podem ser encontradas no nosso sistema: *Coreografia* e *Orquestra*

1.3.1 Coreografia

Este tipo de arquitetura é usado nas comunicações entre:

- ***Monitoring Service - Crosswalk Service***
- ***Monitoring Service - Vehicle Service***
- ***Monitoring Service - Pedestrian Service***

Este estilo de arquitetura assenta nos dois eventos que o *Monitoring Service* publica. O *Crosswalk Service* vai subscrever esses dois eventos sendo que quando ativados este vai realizar tarefas. Assim como o *Pedestrian Service* subscreve o evento relacionado com peões e o *Vehicle Service* o evento direcionado aos veículos.

1.3.2 Orquestra

Este tipo de arquitetura é usado nas comunicações entre:

- ***Crosswalk Service - Notifications Service***

1.4 Colaboração

Na colaboração temos também dois tipos diferentes: *baseado em Eventos* e *request/response*

1.4.1 Baseado em Eventos

Este tipo de colaboração pode ser encontrado:

- ***Monitoring Service - Crosswalk Service***
- ***Monitoring Service - Pedestrian Service***
- ***Monitoring Service - Vehicle Service***
- ***Distance Service - Monitoring Service***

1.4.2 Request/Response

Este tipo de colaboração pode ser encontrado:

- ***Distance Service - Monitoring Service***
- ***Crosswalk Service - Notification Service***