



Universidade de São Paulo
Escola Politécnica

Franco Paes Leme Franco
Guilherme Keller Franco

**Aplicação *WEB* voltada para o gerenciamento e
controle do trânsito**

São Paulo-SP, Brasil

2014

Franco Paes Leme Franco
Guilherme Keller Franco

Aplicação *WEB* voltada para o gerenciamento e controle do trânsito

Trabalho de conclusão de curso apresentado
à Escola Politécnica da Universidade de São
Paulo para obtenção do diploma de Enge-
nheiro Mecatrônico

Escola Politécnica da USP – EPUSP

Engenharia Mecatrônica

Graduação

Orientador: Prof. Dr. Fabrício Junqueira

São Paulo-SP, Brasil

2014

Catálogo-na-publicação

**Franco, Franco Paes Leme
Guilherme, Franco Keller**

**Aplicação WEB voltada para o gerenciamento e controle do
trânsito / F.P.L. Franco e G.F. Keller. – São Paulo, 2014.
69 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia Mecatrônica e de
Sistemas Mecânicos.**

- 1.Engenharia de tráfego (Gerenciamento; Controle)**
- 2.Internet**

**I.Universidade de São Paulo. Escola Politécnica. Departamento
de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.**

Resumo

O objetivo deste trabalho é projetar e implementar um aplicativo, facilmente acessível via *web*, com a finalidade de permitir o controle e monitoramento de elementos de controle de tráfego na cidade de São Paulo. Para alcançar esse objetivo foi necessário um estudo sobre os principais conceitos de engenharia de tráfego, desde as definições dos elementos formadores do tráfego até os principais métodos de controle de tráfego urbano.

O aplicativo é baseado na API (*Application Programming Interface*) do *Google Maps*. Essa interface se integra ao projeto por meio da linguagem de programação Javascript e permite que os agentes visualizem os elementos de tráfego e os controlem em tempo real no mapa. Outras linguagem usadas na implementação foram a PHP (*Hypertext Preprocessor*) e HTML (*HyperText Markup Language*). O banco de dados MySQL, utilizado para armazenar as informações referentes aos elementos de tráfego, foi estruturado e é gerenciado por meio da ferramenta MySQL Workbench.

Abstract

The goal of this work is to project and implement a computer application, easily accessible on the internet, that allows the control and monitoring of urban traffic elements in the city of São Paulo. To reach this goal it was necessary to perform a study about the main concepts of traffic engineering, from the definition of the concepts and traffic elements to the main urban traffic control methods.

The application is based on the Google Map's API(Application Programming Interface). This interface is connected with the other parts of the application through the javascript programming language. It allows the agents to control and visualize the traffic elements in a map in real time. Some other languages used in the implementation of the software were PHP and HTML. The tool that was used to build and manage the database, which holds all the information about the traffic elements, was the MySQL Workbench.

KEYWORDS: TRAFFIC CONTROL, GOOGLE MAPS API, WEB APPLICATION, SEMAPHORE CONTROL.

Lista de ilustrações

Figura 1 – Congestionamentos em São Paulo, fonte CET-SP	14
Figura 2 – Componentes do UTMC	24
Figura 3 – TRANSYT	27
Figura 4 – Hierarquia do sistema SCOOT	28
Figura 5 – CTA	29
Figura 6 – Metodologia Scrum	32
Figura 7 – Diagrama de casos de uso: usuários comuns	33
Figura 8 – Diagrama de casos de uso: administradores	34
Figura 9 – Diagrama de atividades: visualizar fluxo de tráfego	36
Figura 10 – Diagrama de atividades: inserir semáforos	37
Figura 11 – Diagrama de atividades: editar semáforos	38
Figura 12 – Diagrama de atividades: remover semáforos	39
Figura 13 – Diagrama de atividades: consultar estado dos semáforos	40
Figura 14 – Diagrama de Classes	41
Figura 15 – Diagrama de sequência: <i>login</i>	42
Figura 16 – Diagrama de sequência: inserir equipamentos	43
Figura 17 – Diagrama de sequência: editar equipamentos	43
Figura 18 – Diagrama de sequência: remover equipamentos	44
Figura 19 – Diagrama de sequência: inserir histórico	44
Figura 20 – Diagrama de sequência: inserir notificação	45
Figura 21 – Comunicação HTTP	46
Figura 22 – Funcionamento da ferramenta <i>Symfony</i>	46
Figura 23 – Banco de dados em tabela, MySQL Workbench	48
Figura 24 – Página <i>login</i>	50
Figura 25 – Página visualizar tráfego	50
Figura 26 – Página estatísticas	51
Figura 27 – Página problemas	52
Figura 28 – Página semáforos	52
Figura 29 – Página cruzamentos	53
Figura 30 – Página radares	53
Figura 31 – Página câmeras	54
Figura 32 – Página banco de dados	54
Figura 33 – Exemplo Caso de Uso	65
Figura 34 – Exemplo Diagrama de Atividades	66
Figura 35 – Exemplo Diagrama de Classes	68
Figura 36 – Exemplo Diagrama de Sequência	69

Lista de tabelas

Tabela 1 – Tempos Fixos	25
Tabela 2 – Adaptativos: Seleção de Planos	25
Tabela 3 – Adaptativos: Tempo Real	26

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
CET	Companhia de Engenharia de Tráfego
CFTV	Circurito Fechado de Televisão
CSS	<i>Cascading Style Sheets</i>
CTA	Centrais de Tráfego em Area
DENATRAN	Departamento Nacional de Trânsito
FHWA	<i>Federal Highway Administration</i>
HTML	<i>HyperText Markup Language</i>
ID	Indice de Desempenho
MCV	<i>Model-View-Controller</i>
OOP	Programação Orientada a Objetos
OPAC	<i>Optimized Policies for Adaptive Control</i>
PHP	<i>Hypertext Preprocessor</i>
SCATS	<i>Coordinated Adaptive Traffic System</i>
SCOOT	<i>Split Cycle and Offset Optimization Technique</i>
SQL	<i>Structured Query Language</i>
TRANSYT	<i>Traffic Study Network Tool</i>
UC	<i>Use Case</i>
UML	<i>Unified Modeling Language</i>
UTC	<i>Urban Traffic Control</i>
UTMC	<i>Urban Traffic Management and Control</i>
URL	<i>Uniform Resource Locator</i>
URI	<i>Uniform Resource Identifier</i>

Sumário

	Lista de ilustrações	6
	Lista de tabelas	7
1	INTRODUÇÃO	13
1.1	Objetivo	14
1.2	Motivação	14
1.3	Estrutura do trabalho	15
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Introdução da revisão bibliográfica	17
2.2	Histórico	17
2.3	Fundamentos de engenharia de tráfego e controle semafórico	19
2.3.1	Conceitos Básicos	19
2.3.2	Parâmetros caracterizantes do fluxo de tráfego	20
2.3.3	Variáveis de controle semafórico	21
2.4	Metodologias de controle de tráfego	22
2.5	UTC	23
2.5.1	Sistema de tempos fixos	26
2.5.1.1	TRANSYT	26
2.5.2	Sistemas adaptativos	27
2.5.2.1	SCOOT	27
2.6	Sistema de controle semafórico da cidade de São Paulo	28
3	MATERIAIS E MÉTODOS	31
3.1	Metodologia	31
3.2	Requisitos	32
3.2.1	Diagramas	33
3.2.1.1	Diagramas de casos de usos	33
3.2.1.2	Diagramas de atividades	35
3.2.1.3	Diagramas de classes	41
3.2.1.4	Diagramas de sequência	42
3.3	Ferramentas utilizadas	45
3.3.1	Servidor: Apache	45
3.3.2	Symfony	45
3.3.3	FOS User	47

3.3.4	MySQL Workbench	47
4	RESULTADOS	49
4.1	<i>Layout</i>	49
4.1.1	Página: <i>login</i>	49
4.1.2	Página: visualização trânsito	50
4.1.3	Página: estatísticas	51
4.1.4	Página: problemas	51
4.1.5	Página: semáforos	52
4.1.6	Página: cruzamentos	53
4.1.7	Página: radares	53
4.1.8	Página: câmeras	53
4.1.9	Página: banco de dados	54
5	DISCUSSÃO	55
6	CONCLUSÕES	57
6.1	Trabalhos futuros	57
7	REFERÊNCIAS BIBLIOGRÁFICAS	59
	ANEXOS	61
	ANEXO A – UML	63
A.1	Introdução	63
A.2	Caso de Uso	64
A.2.1	Como é um Caso de Uso	65
A.3	Diagrama de Atividades	66
A.3.1	Como é Diagrama de Atividades	66
A.4	Diagrama de Classes	67
A.4.1	Como é um Diagrama de Classes	67
A.5	Diagrama de Sequência	68
A.5.1	Como é um diagrama de Sequência	68

1 Introdução

Problemas relacionados ao fluxo de tráfego não são tão recentes e além disso, soluções para esses problemas foram propostos com o decorrer dos anos de acordo com as necessidades de cada período. O primeiro dispositivo de controle de sinais à luzes foi instalado em Londres no final do século XIX. No entanto, era um sistema simples formado por luzes vermelhas e verdes que dirigiam o tráfego simplesmente em duas ações; ir e parar. (GORDON M.,1971)

Sempre observou-se que o fluxo de veículos poderia ser mais organizado e mesmo novos sistemas poderiam ser implementados para evitar acidentes. Por exemplo, em Salt Lake City, um primeiro sistema foi instalado e era capaz de controlar seis cruzamentos, sendo operado manualmente. A evolução desses sistemas foi rápida chegando a um marco importante que foi o controle por meio de computadores com acesso aos sinais de tráfego em Toronto, Ontario Canada. Desde meados do século XX o funcionamento dos semáforos requer um controle cada vez mais sofisticado para garantir uma coordenação de fluxo e garantir maior segurança. Existem diversos tipos de controle, desde simples contagem de tempo até controle complexo e por meio de computadores que automaticamente conseguem se adaptar ao tráfego em tempo real permitindo o maior fluxo possível e com intuito de evitar congestionamentos e tempo de espera (BARNHART; LAPORTE, 2006).

É importante notar que a evolução dos sistemas de controle de tráfego ocorreu de acordo com as necessidades e problemas que aconteciam devido ao fluxo de veículos e pessoas. Os grandes centros urbanos apresentam graves problemas no tráfego, principalmente relacionados ao tempo de espera devido a congestionamentos como consequência do número de veículos em circulação. De acordo com um levantamento feito pelo Ibope em 2008, 63 por cento dos paulistanos gastam entre 30 minutos e 3 horas nos deslocamentos para escolas, universidades ou trabalho. Além disso, existem um número considerável de automóveis na cidade de São Paulo. Segundo o Departamento Nacional de Trânsito (DENATRAN), em setembro de 2012 a cidade de São Paulo contava com mais de 4,8 milhões de automóveis nas ruas. Porém, o problema não consiste apenas no número de automóveis mas também no sistema de controle em São Paulo e sua infraestrutura. A Figura 1 mostra uma estimativa dos congestionamentos em São Paulo diariamente e dessa forma pode-se observar o caos e os gastos que a cidade de São Paulo apresenta todos os dias em relação ao trânsito.

A secretaria de Estado dos Transportes Metropolitanos estima que as perdas financeiras devido aos problemas enfrentados na cidade de São Paulo como acidentes,

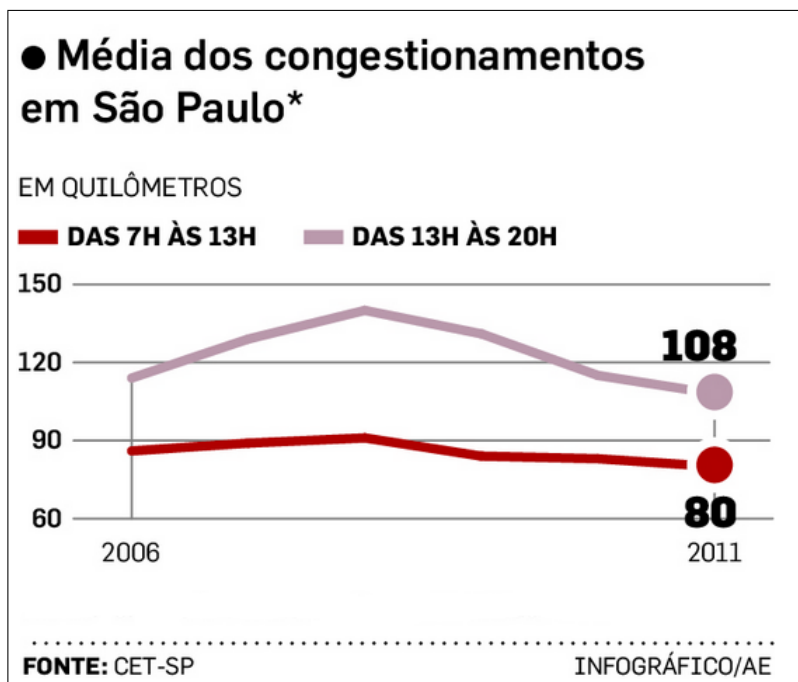


Figura 1 – Congestionamentos em São Paulo, fonte CET-SP

poluição e congestionamentos sejam aproximadamente 4,1 bilhões de reais por ano. Outro estudo realizado pelo Instituto de Estudos Avançados da Universidade de São Paulo estimou que as perdas diárias chegam a 11 milhões de reais contando com o tempo e o combustível nos congestionamentos. Esse resultado foi obtido de acordo com a média de 80 quilômetros de lentidão por dia com picos de 200 quilômetros. Portanto, é importante notar a relevância dos sistemas de controle e sua acessibilidade para amenizar o caos que a cidade de São Paulo enfrenta (CET-SP).

1.1 Objetivo

O trabalho tem como o objetivo o desenvolvimento de um *software* em plataforma web com a função de mapeamento, monitoramento e principalmente a teleoperação de sinais de trânsito na cidade de São Paulo.

Para tanto, deve-se desenvolver uma aplicação web baseada na API do Google Maps facilmente acessível através da *Internet* e projetar um sistema de controle baseado em informações em tempo real sobre o trânsito.

1.2 Motivação

Visto que o trânsito de São Paulo apresenta diariamente problemas que envolvem enormes gastos devido à poluição, congestionamentos e acidentes, o tema deste projeto foi escolhido para gerar futuras possibilidades em um melhor controle do trânsito da cidade

de São Paulo e mesmo em outras cidades brasileiras. Dessa forma, será projetado um *software* em plataforma *web* para que agentes de trânsitos possam ter acesso ao controle do trânsito de São Paulo, contando com diversas opções como o registro de novos semáforos, monitoramento dos semáforos instalados, substituição etc.

1.3 Estrutura do trabalho

Esta monografia está dividida em três partes principais: revisão bibliográfica (seção 2), materiais e métodos (seção 3), resultados (seção 4), discussão (seção 5) e conclusões (seção 6). Na revisão bibliográfica é descrito de forma geral o processo histórico à respeito do sistema de controle de trânsito e os fundamentos da engenharia de tráfego. Já na seção materiais e métodos, as principais ferramentas que foram são utilizadas para alcançar o objetivo são descritas. Na seção 4, tem se o resultado final do projeto, mostrando todas as páginas e *layouts* da plataforma *web*. Por fim, nas seções 5 e 6 são apresentados discussões e as conclusões sobre o projeto.

2 Revisão Bibliográfica

2.1 Introdução da revisão bibliográfica

De acordo com o objetivo deste trabalho, foi necessário inicialmente realizar pesquisas para adquirir um conhecimento e entendimento do controle e monitoramento do tráfego para posterior implementação do aplicativo *web*. Dessa forma, inicialmente esta seção descreverá um pouco mais sobre o histórico dos sistemas de controle de trânsito e posteriormente, os fundamentos de engenharia de tráfego e controle semafórico.

2.2 Histórico

Os sistemas de controle do trânsito são complexos e formados por equipamentos tecnológicos que permitem um melhor gerenciamento de fluxo nos centros urbanos. Mas é importante observar que houve uma evolução nos sistemas de controle, sendo a Inglaterra e os Estados Unidos, os pioneiros no desenvolvimento de novas tecnologias para o controle do tráfego. Mesmo antes de motores a combustão, havia necessidade de uma organização para manter o fluxo de carroças, pessoas e bondes elétricos evitando acidentes. Assim, policiais de trânsito controlavam o fluxo por meio de gestos para evitar acidentes e ter uma melhor dinâmica nos cruzamentos dos antigos centros urbanos. Com a evolução dos motores à combustão, a quantidade de veículos aumentou em grande escala, surgindo a necessidade de uma forma melhor para controlar o trânsito (HOMBURGUER et al., 1992).

Em 1868, um dos primeiros dispositivos para sinalização de trânsito foi instalado em Londres. Ele foi projetado para ajudar policiais de trânsito, especialmente durante à noite, quando a visibilidade era ruim. Era conhecido como braços de semáforo *semaphore-arm* e tinha lentes vermelhas e verdes iluminadas por gás para visualização à noite. Este foi o primeiro registro do uso das cores vermelho e verde para controlar o tráfego. Já nos Estados Unidos, os primeiros dispositivos para o controle do trânsito foram utilizados no começo do ano de 1909. Esses dispositivos geralmente eram semáforos ou setas para direcionar o sentido do trânsito, porém alguns deles já utilizavam algum tipo de fonte de iluminação e sistemas de lentes que também eram usados nos sistemas de ferrovias. Em 1914, em Cleveland, Ohio, instalou-se sinais de duas cores para o controle do tráfego. Tal como era o sinal a gás de Londres. Este dispositivo foi projetado para ajudar a polícia de trânsito em cruzamentos movimentados. Assim, o policial era o responsável em controlar o tráfego por meio desse dispositivo (WEBSTER E COBBE, 1966).

Atingindo o objetivo de controlar localmente um cruzamento, o problema passou

a ser então como interligar vários cruzamentos para permitir um melhor fluxo. Então, em 1917 um primeiro dispositivo foi instalado em Salt Lake City e permitia o controle manual dos semáforos em seis cruzamentos. Outro importante invento foi os semáforos que funcionavam sem a ajuda de policiais, ou seja, eram automáticos. Esse dispositivo foi instalado na Grã Bretanha em Wolverhampton em 1926 e eram semáforos elétricos controlados em tempos fixos (seção 2.5.1). Portanto esse sistema passou a ser cada vez mais utilizado devido a sua simplicidade. Porém o sistema de tempo fixo não era flexível em relação ao tráfego. Os tempos de sinalização eram previstos de acordo com os horários. O problema é que a previsão de tráfego não é tão simples e necessita de maiores coletas de informações para estimar os tempos fixos. Assim, verificava que mesmo sendo um sistema mais simples, esse sistema não era tão eficaz (CLOWES, 1980).

Com a concentração de veículos nos centros urbanos, a evolução dos sistemas de controle do trânsito foi necessária. Com a utilização dos semáforos controlados em tempos fixos observou-se que apesar de um controle automático, o sistema não era tão eficiente em cruzamentos onde haviam grandes fluxos de veículos e onde os fluxos variavam muito no decorrer do dia. Dessa forma, a necessidade de uma melhor programação desses sistemas foi inevitável para obter um melhor controle do tráfego em diferentes horários. Para isso foi necessário também introduzir sistemas mais computadorizados que permitiam um sensoramento e processamento de dados para garantir um melhor controle, nos sistemas chamados *open loops* (seção 2.5.2) (CLOWES, 1980).

Entre 1928 e 1930 foram introduzidos controladores de sistemas atuados por meio de sensores de pressão. Esse sistema coleta informação em tempo real e permite controlar o fluxo de veículos de acordo com as necessidades nos cruzamentos. Em 1952, em Dever Colorado, foi instalado um sistema de controle analógico e seu princípio era um sistema atuado (seção 2.4). Nesse sistema havia sensores que amostravam sinais de fluxo de veículos e adequavam o tempo dos sinais de acordo com as informações guardadas. Já em 1960, em Toronto foi realizado um estudo usando computador digital para executar as funções do sistemas de controle de tráfego. Com isso, maiores informações poderiam ser armazenadas do tráfego em tempo real. Esse sistema, porém, foi essencial para o sucesso desta aplicação. Assim, inicialmente esse sistema foi instalado em 20 cruzamentos sob o controle de um computador. Posteriormente, expandiu-se o sistema para 885 cruzamentos em 1973 (MCSHANE, 1999).

Com a evolução dos sistemas computacionais principalmente devido à IBM foi possível uma melhoria nos sistemas de controle, como por exemplo redução significativa de paradas, atrasos e acidentes. Dessa forma ocorreu uma expansão da aplicação desse tipo de sistema por outras diversas regiões, principalmente, cidades nos EUA. Além disso os sinais de trânsito começaram a ser controlados usando planos de tempo que eram armazenados e desenvolvidos em modo *off-line*. Um marco importante no controle de

semáforos foi em 1967 no qual a *Federal Highway Administration*(FHWA) iniciou seu projeto de desenvolvimento de sistemas de controle de tráfego urbano (UTCS seção 2.5). Era um novo sistema com novas estratégias avançadas para o controle de tráfego. O projeto foi instalado em Washington DC no qual continha 512 detectores de veículos e determinavam o tempo do sinal em 113 cruzamentos. Dessa forma, o sistema tinha mais informações sobre o tráfego em tempo real e esses dados serviam para a investigação estratégica no controle de tráfego. Além disso, as informações serviram de base para o desenvolvimento de novos *softwares*.

Na década de 70, houve desenvolvimento de novos *softwares* ligados ao sistema de controle de tráfego. Com novos sistemas, *Split Cycle and Offset Optimization Technique*(SCOOT seção 2.5.2.1) e *Coordinated Adaptive Traffic System* (SCATS) iniciaram o desenvolvimento de controle sensíveis. Ou seja, eles usavam técnicas de controle adaptativo por meio de estratégias otimizadas para o controle adaptativo conhecido com *Optimized Policies for Adaptive Control* (OPAC).

2.3 Fundamentos de engenharia de tráfego e controle semafórico

2.3.1 Conceitos Básicos

A seguir são apresentados alguns conceitos básicos relativos à engenharia de tráfego necessários para o entendimento de futuras seções do trabalho. Os conceitos apresentados por Gordon e Tighe (2005) e o Departamento nacional de trânsito (1984) são:

- Controle de interseção isolada – controla o tráfego sem considerar interseções adjacentes.
- Controle de cruzamentos e interseções próximas – garante fluxo de tráfego progressivo através de duas interseções próximas, como um cruzamento por exemplo.
- Controle de interseção arterial (rede aberta) – garante fluxo de tráfego progressivo pela via arterial do controle de semáforos.
- Controle de rede fechada – coordena um grupo de interseções adjacentes.
- Sistema de controle de área ampla – trata todos ou grande parte dos sinais de uma cidade ou área metropolitana como um único sistema.
- Movimento – identifica a origem e o destino de veículos e pedestres.
- Movimentos conflitantes - movimentos que se cruzam em uma interseção.
- Aproximação – trechos de via que convergem para a interseção.

- Atraso veicular – demora total causada ao veículo devido ao tempo em vermelho. Leva em conta o tempo perdido na aceleração e na desaceleração.
- Capacidade da interseção – número máximo de veículos que conseguem atravessar a interseção em um período de tempo.
- Defasagem – a diferença de tempo entre os instantes de início de um intervalo em uma interseção e o início desse mesmo estágio em uma outra via de referência.

2.3.2 Parâmetros caracterizantes do fluxo de tráfego

Segundo Roess; Prassas e McShane (2004), os parâmetros que caracterizam o fluxo de tráfego estão contidos em duas grandes categorias, os parâmetros macroscópicos e os microscópicos. Parâmetros macroscópicos descrevem o fluxo de veículos no sistema como um todo, enquanto que os parâmetros microscópicos descrevem o comportamento de um ou dois carros dentro do fluxo. Os três principais parâmetros macroscópicos são:

- Fluxo ou volume – é definido como o número de veículos passando por um certo ponto de uma pista ou uma via em uma certa direção em um intervalo de tempo preestabelecido. Tem como unidade de medida “veículos por unidade de tempo”, sendo que as unidades de tempo mais usadas são uma hora ou um dia. O fluxo é geralmente expresso em veículos por hora mesmo quando o período observado é menor do que uma hora, um fluxo de 200 veículos em 15 minutos equivale à 800 veículos por hora em um intervalo de quinze minutos. A expressão que descreve o fluxo é mostrada abaixo:

$$q = \frac{N}{T}$$

Onde N representa o número de veículos e T a unidade de tempo.

- Velocidade e tempo de atravessamento – "velocidade" é definida como distância percorrida por unidade de tempo, enquanto que "tempo de atravessamento" é definido como o tempo necessário para atravessar uma certa seção da via. Tempo de atravessamento é o inverso da velocidade. Em um fluxo de tráfego cada veículo tem uma velocidade diferente, o parâmetro macroscópico é, portanto, caracterizado pela velocidade média dos veículos nesse fluxo. Existem dois modos em que a velocidade média de um fluxo pode ser medida:

o Velocidade média temporal – refere-se à velocidade média de todos os veículos passando por um certo ponto da via em um período de tempo específico.

o Velocidade média espacial – refere-se à velocidade média de todos os veículos ocupando um trecho da via em um periodo de tempo específico.

- Densidade e ocupância – "densidade" é definida como o número de veículos ocupando um certo trecho da via e é geralmente expressa em veículos por quilômetro. A densidade é difícil de ser medida diretamente, já que exige uma visão ampla de um trecho de estrada, portanto ela é normalmente obtida através de medidas de velocidade e fluxo usando a fórmula abaixo:

$$D = \frac{v}{S}$$

Onde v corresponde ao fluxo (veículos/hora) e S corresponde à velocidade média espacial (quilômetros/hora). A ocupância, diferentemente da densidade, pode ser facilmente medida de forma direta por meio de sensores nas pistas. Ela é definida como a porcentagem de tempo na qual um detector é ocupado por um veículo em um dado intervalo de tempo. A densidade pode ser obtida a partir da ocupância de acordo com a seguinte expressão:

$$D = \frac{O}{Lv + Ld}$$

Onde O é a ocupância, Lv o comprimento médio de um veículo e Ld o comprimento do detector.

Pode-se ainda destacar os parâmetros microscópicos que são:

- Espaçamento – é definido como o espaço entre dois veículos sucessivos na mesma faixa. É medido entre dois pontos iguais dos veículos, como a roda da frente por exemplo.
- *Headway* – é definido como o tempo entre a passagem de dois veículos sucessivos por um ponto da via.

2.3.3 Variáveis de controle semafórico

Quando propriamente instalados e controlados, os semáforos podem proporcionar uma série de benefícios como: aumentar a capacidade da interseção, diminuir a ocorrência e severidade de certos tipos de acidentes, prover um fluxo praticamente contínuo de veículos em vias arteriais (ROESS; PRASSAS; MCSHANE, 2004).

A seguir são apresentados alguns conceitos básicos usados no projeto de temporização de semáforos segundo Roess; Prassas e McShane (2004) e Departamento Nacional de Trânsito (1984):

- Ciclo – refere-se a uma rotação completa através de toda a sequência de sinalização em uma interseção.
- Duração do ciclo – tempo total para completar a sequência de sinalização em uma interseção. É medido em segundos.
- Intervalo – é o intervalo de tempo no qual não há mudança de estado do semáforo. Existem diferentes tipos de intervalos contidos em um ciclo.

o Intervalo de transição – equivale ao estado “amarelo” do semáforo. É parte da transição do estado “verde” para “vermelho”, na qual o semáforo prestes a deixar o estado verde muda para amarelo enquanto os outros permanecem no estado vermelho. É temporizado de forma a permitir que um veículo, que não consiga parar imediatamente ao fim do estado verde, atravesse a interseção de forma segura.

o Intervalo de desobstrução – também faz parte da transição de verde para vermelho. Durante esse intervalo todos os semáforos estão no estado vermelho. Ele tem a função de permitir que os movimentos que entrem na interseção durante o estado amarelo tenham tempo de deixar a mesma antes que o tráfego de outros movimentos concorrentes seja liberado.

o Intervalo verde – refere-se ao intervalo em que o semáforo em questão está no estado verde, liberando o movimento, enquanto todos os outros estão no estado vermelho.

o Intervalo vermelho – refere-se ao estado em que o semáforo está no estado vermelho, impedindo o movimento, enquanto algum outro semáforo está no estado verde.

- Fase – é a sequência verde, amarelo e vermelho aplicada a um ou mais movimentos.

2.4 Metodologias de controle de tráfego

Segundo Roess; Prassas e McShane (2004), diferentes métodos podem ser usados para controlar os semáforos. Existe o controle de tempo fixo, o controle semi ou comple-

tamente atuado por veículos detectados por sensores. A escolha do modelo depende de variáveis como disponibilidade de infraestrutura de comunicação, características do fluxo de veículos no local, espaçamento da interseção e custo de instalação e manutenção dos detectores (PIAI, 2009).

- Controle em tempo fixo - em uma operação em tempo fixo o ciclo, a sequência de fases e a temporização de cada intervalo são constantes. Cada ciclo do semáforo segue um plano predeterminado, podendo haver diferentes planos usados para diferentes horários do dia para lidar com os horários de pico.
- Controle semi atuado - Na operação semi atuada detectores de veículos são instalados nas vias secundárias que dão acesso à via principal, não existem detectores na via principal. O sinal fica verde na via principal até que um veículo seja detectado na via secundária. Então, de acordo com as regras de tempo mínimo e máximo de verde na via principal, o sinal verde é transferido para a via secundária. O verde retorna para a via principal quando o tempo máximo de verde é atingido na via secundária ou quando os sensores param de detectar demanda de veículos nessa via.
- Controle completamente atuado - nesse tipo de controle todas as faixas de todas as vias de um interseção devem ser monitoradas por detectores. O Tempo de verde é alocado de acordo com informações provenientes dos detectores e regras preestabelecidas nos controladores dos semáforos. No controle completamente atuado a duração do ciclo, o tempo de verde e a sequência de fases podem variar de ciclo para ciclo.

Nos sistemas atuados as informações sobre o tráfego são obtidas através de detectores que podem ser detectores de laço indutivo enterrados sob a superfície da via ou detectores de microondas instalados em postes. O Controlador dos semáforos pode estender o tempo de verde de um semáforo desde um mínimo até um máximo prestabelecidos (SLINN; MATTHEWS; GUEST, 2005).

2.5 UTC

O grande volume de veículos e a complexidade do trânsito nos grandes centros urbanos torna necessário o uso de sistemas ótimos de controle e gerenciamento de trânsito para se obter um sistema de trânsito eficiente, seguro e sustentável. Alguns fatores exigidos pelos usuários do sistema que o tornam ainda mais complexo são a priorização do transporte público, informação de tráfego em tempo real, gerenciamento de acidentes e emergências, restrição de tráfego em áreas sensíveis, entre outros. Como resposta a esses problemas foram introduzidas em várias cidades ao redor do mundo os sistemas chamados UTMC (*Urban Traffic Management and Control*). Esses sistemas visam proporcionar

ferramentas para auxiliar no gerenciamento eficiente de redes de tráfego urbano tendo em vista problemas de tráfego atuais e possíveis problemas futuros (BARNHART; LAPORTE, 2006).

Os sistemas UTMC englobam e expandem funcionalidades de outros sistemas de controle de tráfego já existentes como mostrado na Figura 2 . Normalmente os sistemas UTC (*Urban Traffic Control*) são o centro dos UTMC. UTC se refere ao controle de tráfego em áreas urbana através do uso de semáforos conectados entre si para operar em um modo coordenado. Tais sistemas podem ser utilizados para tornar realidade uma série de metas de tráfego eficiente como melhoria da segurança, redução de poluição, prioridade para certos grupos de usuários do sistema de tráfego e mitigação dos efeitos de eventos como acidentes e fechamento de vias (BARNHART; LAPORTE, 2006).

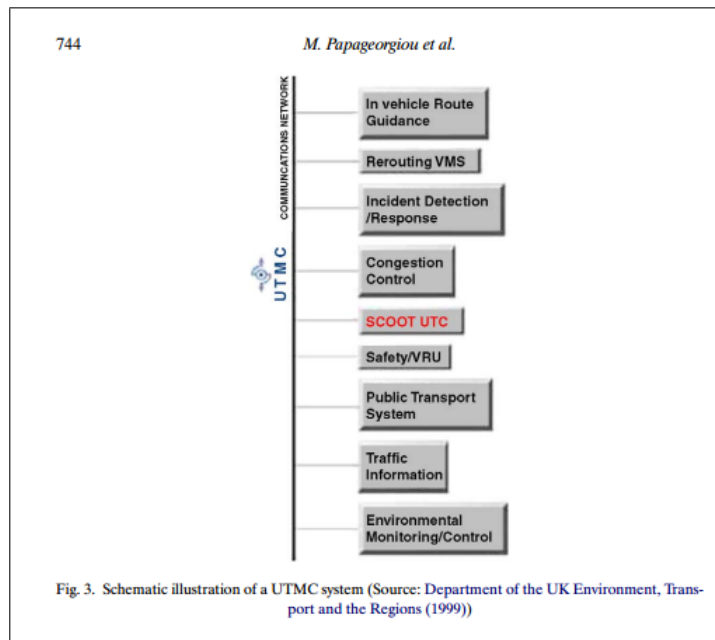


Figura 2 – Componentes do UTMC

Sistemas UTC podem ser baseados em informações históricas ou em tempo real da rede de tráfego. Os sistemas que se baseiam em informações históricas são os sistemas de tempo fixo e os sistemas que se baseiam em informações em tempo real geradas por sensores nas vias são chamados de sistemas adaptativos. Os sistemas adaptativos podem ainda ser divididos em sistemas de seleção de planos ou sistemas adaptativos em tempo real. Nos sistemas com seleção de planos alguns planos de tempo fixo pré programados são selecionados de acordo com a demanda de tráfego medida pelos detectores. Os sistemas adaptativos em tempo real calculam continuamente os parâmetros ótimos de controle (SLINN; MATTHEWS; GUEST, 2005). As Tabelas 1, 2 e 3 listam as principais vantagens e desvantagens de cada um desses sistemas.

Tabela 1 – Tempos Fixos

Vantagens	Desvantagens
<ul style="list-style-type: none"> -Baixo custo de instalação e manutenção -Pode ser implementado usando equipamento de controle não centralizado -Familiaridade para programação pelos técnicos do órgão -Ondas verdes são mais facilmente implementáveis -Pode priorizar tipos de veículos específicos mais facilmente 	<ul style="list-style-type: none"> -Os planos semafórico necessitam de revisões periódicas -Ruptura de coordenação nas trocas de planos semafóricos -Necessita de atuação do operador em casos de incidentes e variações de fluxo imprevistas -Não permite tratar pequenas flutuações de tempos para níveis de fluxo

Tabela 2 – Adaptativos: Seleção de Planos

Vantagens	Desvantagens
<ul style="list-style-type: none"> -Permite tratar as variações diárias de níveis de fluxo -Os horários de troca dos planos são mais apropriados -Familiaridade para programação pelos técnicos do órgão -Possui custo mais baixo que o sistema adaptativo em tempo real, pois necessita de menos detetores 	<ul style="list-style-type: none"> -Necessita de muito mais dados para serem coletados do que o sistema de tempo fixo -Possibilidade de falhas nos detetores -Ruptura de coordenação nas trocas de planos semafóricos -Pode trocar de plano por uma decisão errada -Dificuldade de prever todos os planos que serão necessários

Tabela 3 – Adaptativos: Tempo Real

Vantagens	Desvantagens
-Poucos dados necessários para serem coletados	-Possibilidade de falhas nos detetores, a qual comprometeria o desempenho do sistema
-Resolve os problemas relativos à trocas e revisões dos planos semafórico	-Elevado custo de implantação e manutenção
-Permite tratar com pequenas e grandes variações os diferentes níveis de fluxo	-É imprescindível a central de controle, sem a qual os controladores não operam de forma viável
-Atuação automática frente à incidentes	-A manutenção é um item crítico
-Monitora a situação do tráfego em toda área	

2.5.1 Sistema de tempos fixos

Em sistemas de tempo fixo a otimização dos parâmetros de controle é feita de forma *off-line* baseada em níveis de demanda que são considerados constantes no período em que cada plano de tempo fixo deve funcionar (BARNHART; LAPORTE, 2006).

Sistemas de tempo fixo podem ser facilmente usados para criar ondas verdes, prioridade para certos tipos de veículos ou lidar com eventos especiais que podem ser previstos, como uma partida de futebol. Eles não podem, porém, lidar com problemas imprevisíveis como acidentes ou trabalho nas vias. As mudanças de planos podem ser ativadas por horários pre determinados ou medidas de filas e tráfego em localizações pré determinadas (BARNHART; LAPORTE, 2006).

A seguir é apresentado um exemplo de sistema de tempo fixo, o TRANSYT.

2.5.1.1 TRANSYT

TRANSYT (*Traffic Study Network Tool*) é um Sistema *off-line* para cálculo de temporizações otimas em redes de semáforos. Este é o sistema mais utilizado mundialmente no controle de semáforos em tempo fixo. Ele assume que o fluxo médio de veículos em uma via em um certo período de tempo é conhecido e constante (BARNHART; LAPORTE, 2006).

O sistema é composto por 2 elementos principais, o modelo de tráfego e o procedimento de otimização como mostrado na Figura 3 (PIAI, 2009).

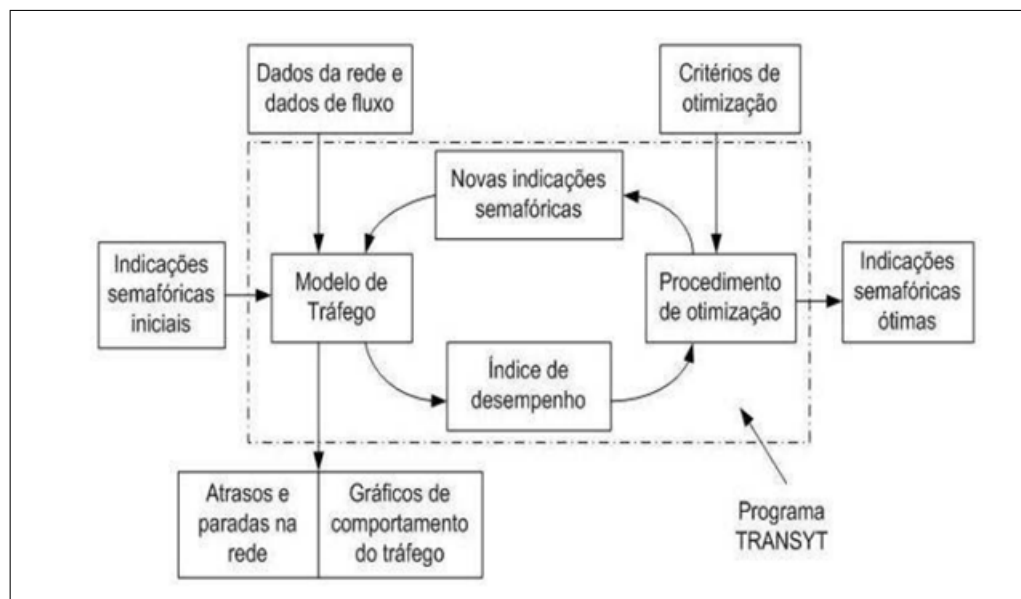


Figura 3 – TRANSYT

O modelo de tráfego representa o comportamento do tráfego em um rede de vias e prediz um “índice de desempenho” (ID) para a rede para um dado plano de tempo fixo e um dado conjunto de fluxos de veículos. O ID mede o custo geral de congestionamento de tráfego, que é geralmente uma combinação ponderada do atraso total e do número de paradas feitas pelos veículos.

2.5.2 Sistemas adaptativos

Os sistemas adaptativos com seleção de planos se utilizam de dados do tráfego obtido através de detectores para selecionar o plano pre programado mais apropriado para as características do fluxo atual. Os planos pre programados devem ser gerados usando uma ferramenta como o TRANSYT (SLINN; MATTHEWS; GUEST, 2005).

Sistemas adaptativos em tempo real utilizam medidas de detectores *on-line* para otimizar os parâmetros de controle semaforico a cada ciclo. Esse tipo de sistema pode ter diferentes graus de centralização. Alguns sistemas podem ser controlados por um computador central e controladores periféricos mais simples, enquanto outros podem ter controladores inteligentes distribuídos pela malha (BARNHART; LAPORTE, 2006).

A seguir são apresentados alguns exemplos de sistemas adaptativos.

2.5.2.1 SCOOT

O sistema SCOOT (*Split Cycle Offset Optimization Technique*), criado no Reino Unido e utilizado ao redor do mundo, baseia-se em três princípios: o cálculo dos PFC (Perfis de fluxo cíclicos), a atualização a cada 4 segundos dos modelos de tráfego *on-line* em

cada *link* e a otimização incremental dos tempos dos semáforos (BARNHART; LAPORTE, 2006).

O otimizador do semáforo utiliza o modelo de tráfego em tempo real para otimizar o tempo de ciclo, a defasagem e a fração de tempo de verde. Cada um desses três parâmetros utiliza uma técnica e uma frequência de otimização diferentes. Através de pequenas mudanças incrementais na temporização dos semáforos, o sistema consegue responder tanto às flutuações de curto prazo na demanda de tráfego quanto tendências de demanda que se estendem por um longo horizonte temporal (BARNHART; LAPORTE, 2006).

Segundo Ming (1997) a hierarquia do sistema SCOOT é a seguinte:

- Área – toda a região controlada pelo sistema.
- Região – subdivisão de área. É um grupo de semáforos que operam com o mesmo tempo de ciclo e são coordenados entre si por meio de defasagens.
- Grupo – subdivisão de região. Cada nó em um grupo pode ser designado mestre ou escravo. Quando um mestre cai para o modo local todos os outros nós caem com ele. Quando um escravo cai para modo local os outros continuam operando normalmente.
- Nó – é a menor unidade de controle. Corresponde à uma interseção semaforizada.
- *Link* – é o que compõe os nós. Corresponde à uma aproximação do cruzamento semaforizado.

A Figura 4 apresenta a hierarquia do sistema SCOOT:

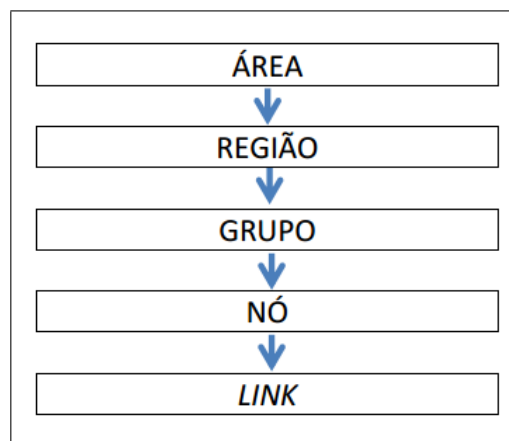


Figura 4 – Hierarquia do sistema SCOOT

2.6 Sistema de controle semafórico da cidade de São Paulo

Desde 1992 a cidade de São Paulo já dispunha de um sistema de controle semafórico centralizado chamado SEMCO. Com exceção de alguns poucos semáforos atuados, a maioria

dos semáforos desse sistemas usava o controle em tempo fixo. Isso gerava uma falta de flexibilidade no sistema que causava um atendimento ineficiente nas regiões com um perfil de tráfego mais volátil e uma incapacidade de lidar com incidentes imprevisíveis nas vias da cidade (VILANOVA, 2005).

Para resolver as ineficiências do sistema SEMCO, entre os anos de 1995 e 1997, foi implementado o Projeto CTA (Centrais de tráfego em área), um sistema centralizado que operava em tempo real. A experiência com o SEMCO demonstrou que um sistema muito centralizado causava ineficiências administrativas e operacionais. Então no novo projeto decidiu-se construir 5 Centrais de tráfego em área, cada uma com a função de controlar o tráfego de uma determinada área da cidade (VILANOVA, 2005). As 5 centrais e as áreas controladas por cada uma são apresentadas a seguir:

- CTA-1 – Centro expandido
- CTA-2 – Oeste e Norte
- CTA-3 - Leste
- CTA-4 - Sudeste
- CTA-5 – Sul

Em cada um dos 5 centros de controle os semáforos foram agrupados em conjuntos de em média 10 cruzamentos cada, chamados de "regiões". Cada uma dessas regiões deveria necessariamente operar de forma coordenada entre si (VILANOVA, 2005). A Figura 5 mostra o esquema de configuração física de uma CTA.

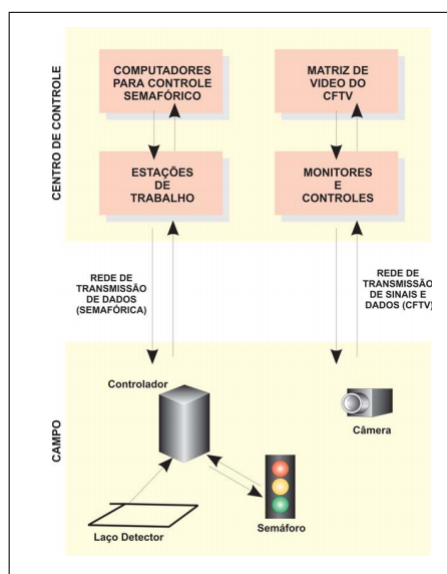


Figura 5 – CTA

As CTAs contam não só com equipamentos de controle semafórico mas também com sistemas de CFTV (circuito fechado de televisão) para ajudar no monitoramento de controle do tráfego. Em 2008 as CTAs possuíam 186 câmeras instaladas. Além disso, em 2008 a cidade de São Paulo possuía 5470 interseções semaforizadas, das quais 1449 faziam parte do sistema de controle em tempo real, 2690 possuíam equipamentos eletrônicos com recursos de nível intermediário e apenas 1195 possuíam equipamentos eletrômecânicos, o tipo mais antigo e mais simples de controlador (CHRISTIANINI; WALDEMAR, 2008).

O sistema adaptativo escolhido para operar na cidade de São Paulo foi o sistema inglês SCOOT (VILANOVA, 2005). Para permitir que diferentes controladores de diferentes marcas pudessem ser conectados as centrais de tráfego utilizando protocolos padronizados, a portaria SMT028/2013 foi publicada em 2013 estabelecendo o padrão de comunicação de controladores semafóricos e centrais de tráfego a ser utilizado na cidade de São Paulo. O padrão escolhido foi o UTMC (*Urban Traffic Management Control*) (ANGELO, 2014).

3 Materiais e métodos

Conforme abordado na seção 2, já existe uma infraestrutura para controle de tráfego e semafórico em diversas cidades desde sistemas controlados em tempos fixos à sistemas adaptativos. Na cidade de São Paulo, por meio das CTAs tem um monitoramento de controle de tráfego. A Internet pode ser utilizada para permitir um melhor controle e comunicação entre as CTAs em tempo real. Além disso, uma plataforma web que auxilia no monitoramento via *Internet* tem vantagens principalmente do ponto de vista econômico para sua implementação e manutenção.

Este projeto trata do desenvolvimento de uma plataforma *web* que permita o administrador gerenciar o funcionamento de equipamentos presentes no trânsito (semáforos, painéis eletrônicos, câmeras e radares) por meio da *Internet*. Nesta seção será feita uma descrição dos requisitos necessários para o desenvolvimento do aplicativo *web*. Posteriormente, as soluções são apresentadas por meio dos diagramas de casos de uso, atividades, classes e sequências com suas respectivas descrições. Por fim, as ferramentas utilizadas para o desenvolvimento da plataforma *web* serão descritas.

3.1 Metodologia

A metodologia escolhida para o desenvolvimento do projeto foi a SCRUM, que é uma maneira ágil para a conclusão de projetos. A metodologia Scrum foi inicialmente formalizada para projetos de desenvolvimento de *software*, mas ele funciona bem para qualquer escopo complexo (SUTHERLAND; SCHWABER, 2013). Primeiramente, é necessário criar uma lista de objetivos priorizados, conhecidos como *backlog*. Posteriormente, são escolhidos alguns dos objetivos principais da lista *backlog* conhecidos como *sprint backlog* e assim, durante essa etapa é necessário decidir como implementar os objetivos. Dessa forma define-se também o tempo, conhecido como *sprint*, necessário para a execução de cada item da lista. Ao longo do *sprint* é necessário focalizar no objetivo principal do projeto. Após o término do *sprint* é preciso divulgar o resultado do desenvolvimento e revisá-lo. Assim, o ciclo se reinicia com a nova escolha do *sprint backlog*. A Figura 6 demonstra como é organizada a metodologia Scrum.

Dessa forma, cada *sprint backlog* foi definida de forma bem estruturada priorizando o objetivo principal do projeto de controle e mapeamento semafórico. O tempo de execução da *sprint backlog* foi definido semanalmente. Após o tempo de execução da lista de objetivos, o tempo de revisão foi definido no máximo um dia para reiniciar o ciclo atingindo os outros objetivos da *backlog*.

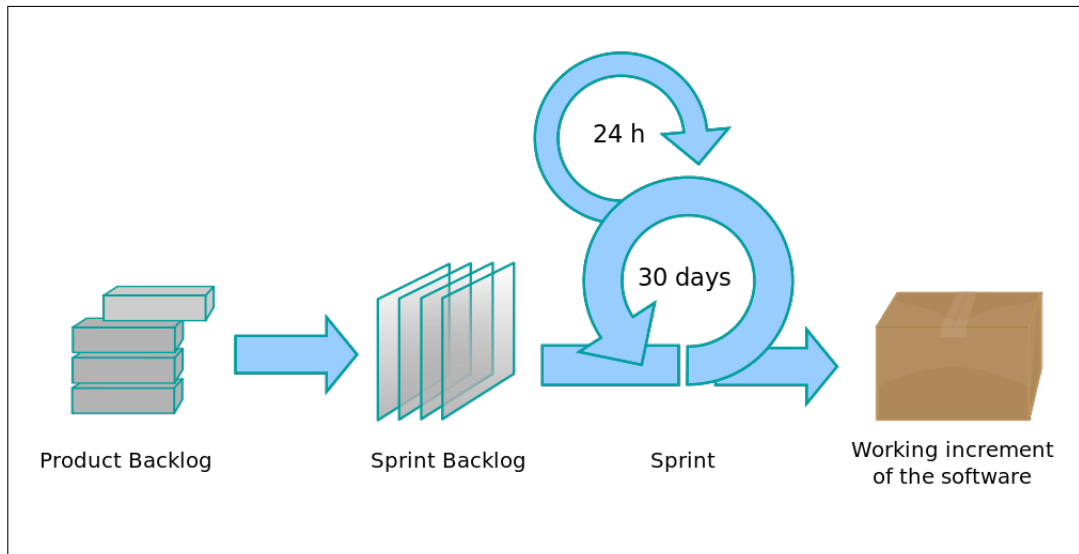


Figura 6 – Metodologia Scrum

3.2 Requisitos

O principal objetivo do projeto em ter uma plataforma web para o controle e mapeamento semaforico é obtido primeiramente com o desenvolvimento do banco de dados para registrar os semáforos e seus dados. Dessa forma, é necessária uma comunicação com o banco de dados em todo instante permitindo assim, adicionar novos equipamentos, editar e mesmo excluir. Portanto, para o gerenciamento de dados (semáforos) é necessário o desenvolvimento do banco de dados para salvar todas as informações que o usuário tem acesso. Além disso, antes de iniciar a programação em si da plataforma web é preciso identificar todas as funcionalidades que o usuário pode realizar.

Sendo um projeto que monitora o tráfego da cidade de São Paulo em tempo real, é fundamental que o site tenha acesso ao mapa com informações da cidade. Para isso é importante utilizar aplicações como, por exemplo, API do Google Maps que disponibiliza informações da cidade e também o fluxo de veículos das principais vias de São Paulo.

Por fim, para o desenvolvimento de um site é preciso utilizar linguagens de programação *web*. Existem diversas linguagens que permitem esse desenvolvimento e nesse projeto utilizou-se principalmente PHP, JavaScript, jQuery, HTML, SQL (*Structure Query Language*), Twig e CSS (*Cascading Style Sheets*). Além dessas linguagens, bibliotecas foram utilizadas para facilitar o desenvolvimento da plataforma como a biblioteca Sonata Project (*SONATA PROJECT, 1998*) com pacotes Symfony (*THE BOOK REFERENCE, 1998*) e Fos User (*FOS USER REFERENCE, 1998*).

3.2.1 Diagramas

Visto que o projeto tem como objetivo desenvolver uma ferramenta em plataforma *web* para o gerenciamento e mapeamento de equipamentos no tráfego em tempo real, é preciso identificar todas as funcionalidades da plataforma e quais e por quem as principais tarefas podem ser executadas. Como solução, foram implementados os diagramas de casos de usos, atividades, classes e sequência descritos a seguir.

3.2.1.1 Diagramas de casos de usos

No projeto decidiu-se que existem dois tipos de atores: administradores e usuários comuns. As ações a serem executadas depende diretamente do papel do ator. Assim, os administradores possuem acesso completo da plataforma *web* para o gerenciamento e os usuários comuns tem um acesso restrito. Dessa forma, as ações podem ser divididas de acordo com os atores. As Figuras 7 e 8 representam os diagramas de casos de usos para os administradores e usuários comuns respectivamente.



Figura 7 – Diagrama de casos de uso: usuários comuns

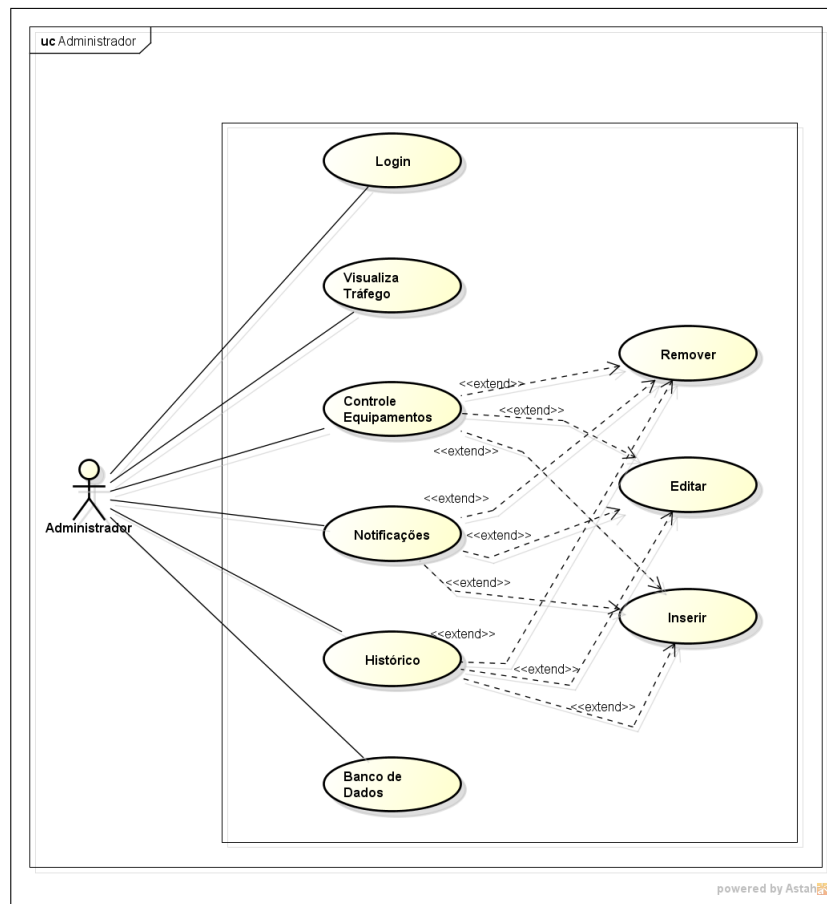


Figura 8 – Diagrama de casos de uso: administradores

Para os usuários comum, as ações principais são:

1. *Login* - quando usuário acessa a página principal da plataforma *web*, é possível visualizar o quadro para inserir seus dados e senha, realizando o *login*.
2. Criar *login* - possibilidade de registrar-se na plataforma *web*. Ao acessar a página principal é possível criar uma conta própria, permitindo aos usuários participar do aplicativo. Ao inserir os dados para cadastro a própria plataforma verifica se existe algum erro na confirmação da senha e *email*.
3. Visualizar tráfego - uma vez feito o *login*, a página principal dos usuários é o mapa com as informações do tráfego da cidade de São Paulo. Por meio do Google Maps, os usuários podem ver informações do tráfego em tempo real. Além disso, é possível buscar o endereço e utilizar as ferramentas do Google Maps como o *Street View*.
4. Visualizar histórico - o aplicativo oferece a opção de salvar informações sobre o tráfego de determinado endereço da cidade de São Paulo por meio dos administradores. Dessa forma, os usuários podem consultar gráficos que informam o histórico do fluxo de veículos em determinado endereço. Essas informações também poderiam ser utilizadas para o controle de semáforos não atuados.

5. Notificações de tráfego - no campo de notificações de problemas os usuários comuns podem inserir informações sobre o trânsito e dessa forma contribuir para um melhor gerenciamento e controle. Para inserir as notificações, os usuários devem selecionar a posição e descrever o possível problema que ocorre em tempo real. Além disso, as notificações feitas pelos administradores são visualizadas nesse mesmo campo.

Para os administradores, as ações principais são:

1. *Login* - como os usuários, os administradores também fazem o *login* para acessar a página principal de gerenciamento.
2. Visualizar tráfego - uma vez feito o *login*, a página é atualizada e o fluxo de tráfego na cidade de São Paulo é acessada.
3. Controle equipamentos - nesta seção é onde os administradores podem gerenciar todos os equipamentos inseridos no mapa da cidade. Assim, os equipamentos são visualizados em sua posição real por meio do Google Maps. Além disso, os administradores podem inserir, editar e remover os equipamentos disponíveis (semáforos, câmeras, radares e painéis eletrônicos) de tal forma que permite manter as informações de todos equipamentos atualizadas.
4. Notificação tráfego - responsável por notificar os usuários comuns pelos problemas do trânsito em tempo real. Portanto, o administrador pode inserir, remover, editar as notificações. Além disso, os administradores podem visualizar as notificações enviadas pelos usuários comuns e ao mesmo tempo gerenciar essas informações.
5. Histórico - por meio das informações do trânsito disponíveis pelo Google Maps, os administradores conseguem salvar os dados e as classificações de trânsito de determinada localidade. Assim com essas informações, os administradores podem visualizar e gerar gráficos do histórico do fluxo do trânsito em determinado endereço de acordo com o dia da semana e horários. Essas informações também poderia ser usadas para o controle de semáforos não atuados.
6. Banco de dados - onde o administrador pode visualizar todo os equipamentos que foram inseridos com opções de edição, exclusão e filtros de busca. É uma seção exclusiva dos administradores e todas as informações podem ser visualizadas nesta seção. Além disso, permite um melhor controle dos equipamentos.

3.2.1.2 Diagramas de atividades

Os diagramas de atividades foram considerados para ter uma melhor representação dos fluxos de atividades que são conduzidos por processamentos. Assim, são apresentados

os principais diagramas de atividades como visualizar fluxo de trânsito, fazer *login*, editar, inserir e remover semáforos. É importante notar que os processos de edição, remoção e inserção dos outros equipamentos são semelhantes aos do semáforos, diferenciando somente nas informações a serem inseridas correspondente a cada equipamento. Outra característica importante é que o estado dos semáforos é exibido automaticamente quando a página de controle semafórico é acessada.

Para visualizar o fluxo de tráfego é necessário clicar na seção trânsito. Assim, a plataforma *web* disponibiliza o mapa com as informações de trânsito e mesmo com a opção de busca de endereço por meio do aplicativo do Google Maps. A Figura 9 demonstra esse processo.

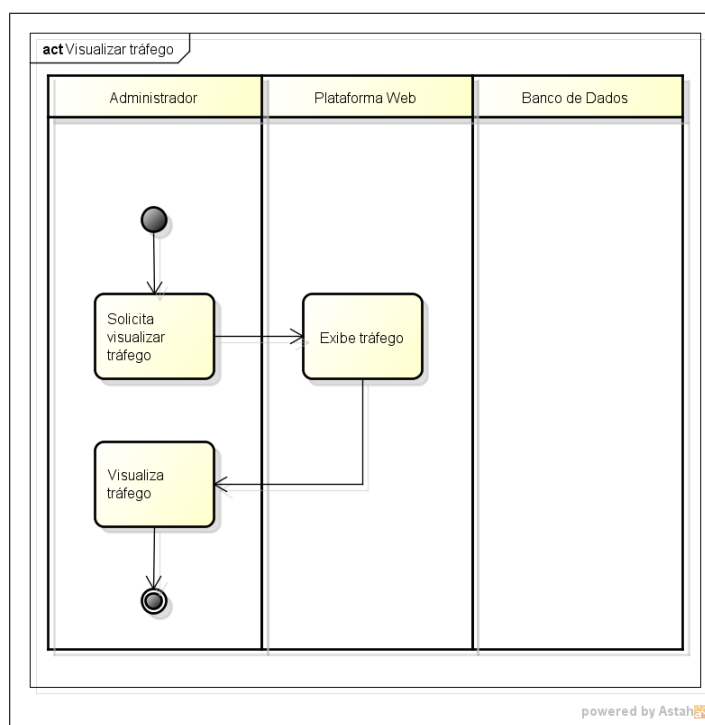


Figura 9 – Diagrama de atividades: visualizar fluxo de tráfego

Os diagramas de atividades do projeto apresentam duas partes principais. A primeira parte refere-se à solicitação do administrador para ter acesso a determinada funcionalidade e consequentemente, a visualização da página correspondente à funcionalidade. A segunda parte refere-se as opções que o administrador pode executar como inserir, editar e remover. A Figura 10 descreve o fluxo de processos de como inserir um semáforo e gravar suas informações no banco de dados. Para isso, o administrador deve clicar na seção semáforos e inserir as informações que a plataforma *web* solicita.

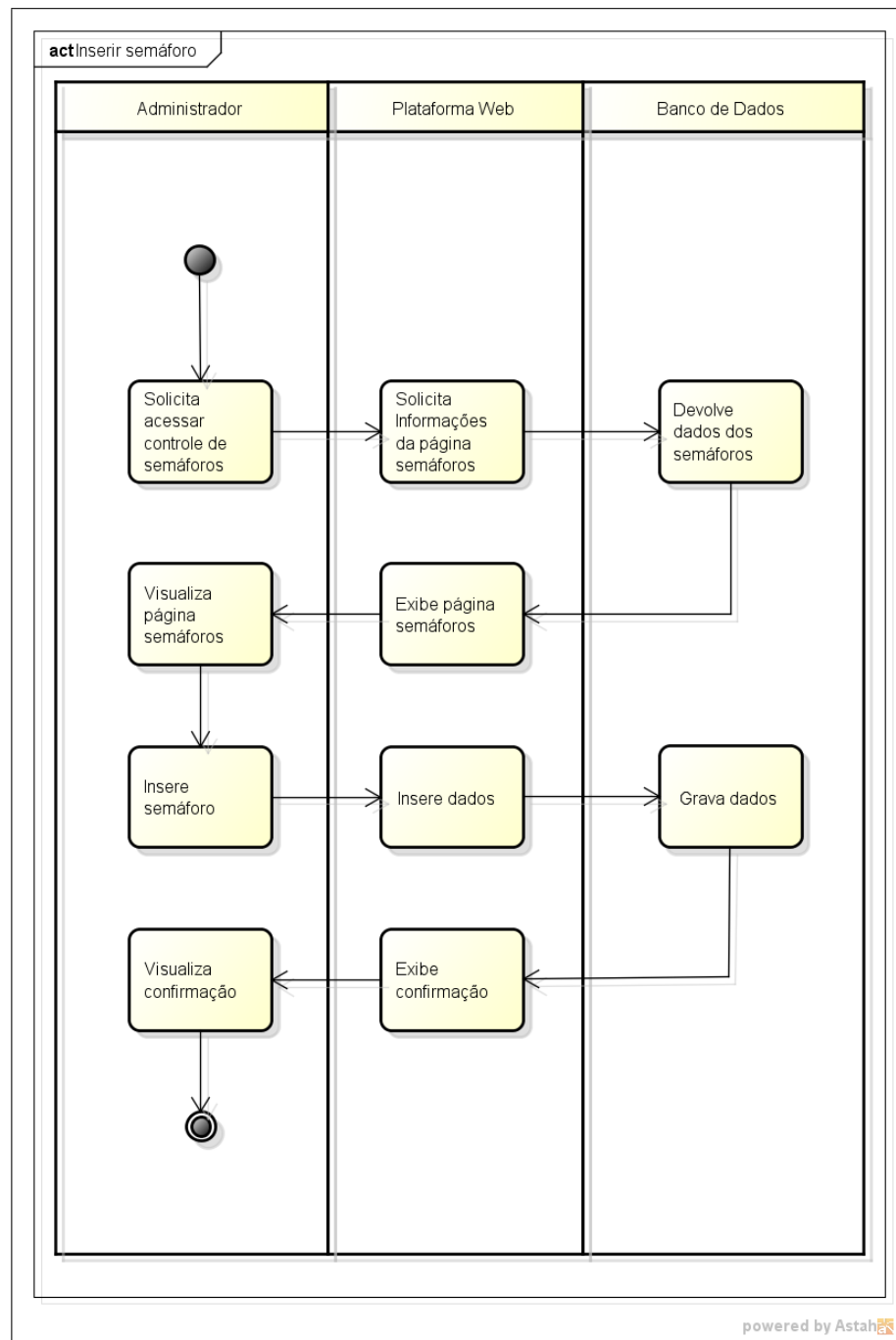


Figura 10 – Diagrama de atividades: inserir semáforos

Uma vez solicitado a seção semáforos, há uma tabela com todos os semáforos inseridos pelos administradores. Para a edição deles é necessário clicar no botão de edição respectivo ao semáforo desejado. Assim, as informações desse semáforo são devolvidas pelo banco de dados e o administrador pode alterar qualquer dado e confirmar a alteração. A Figura 11 representa esse processo.

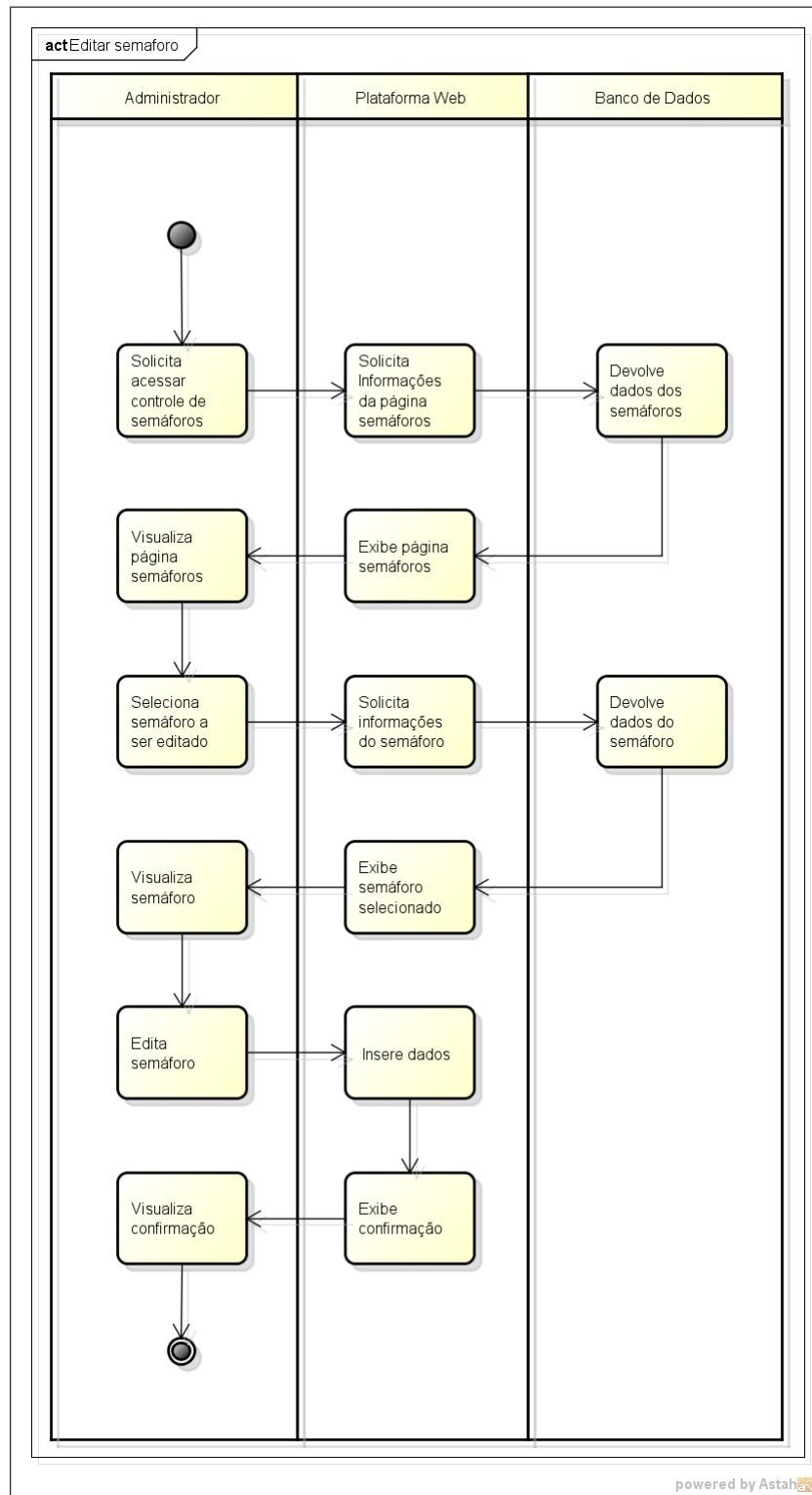


Figura 11 – Diagrama de atividades: editar semáforos

Assim como o processo de edição, a remoção segue as mesmas etapas, diferenciando se apenas o botão específico para remover o semáforo desejado. Após a remoção a plataforma web exibe uma mensagem de confirmação. A Figura 12 ilustra o modelo dinâmico do processo de remoção.

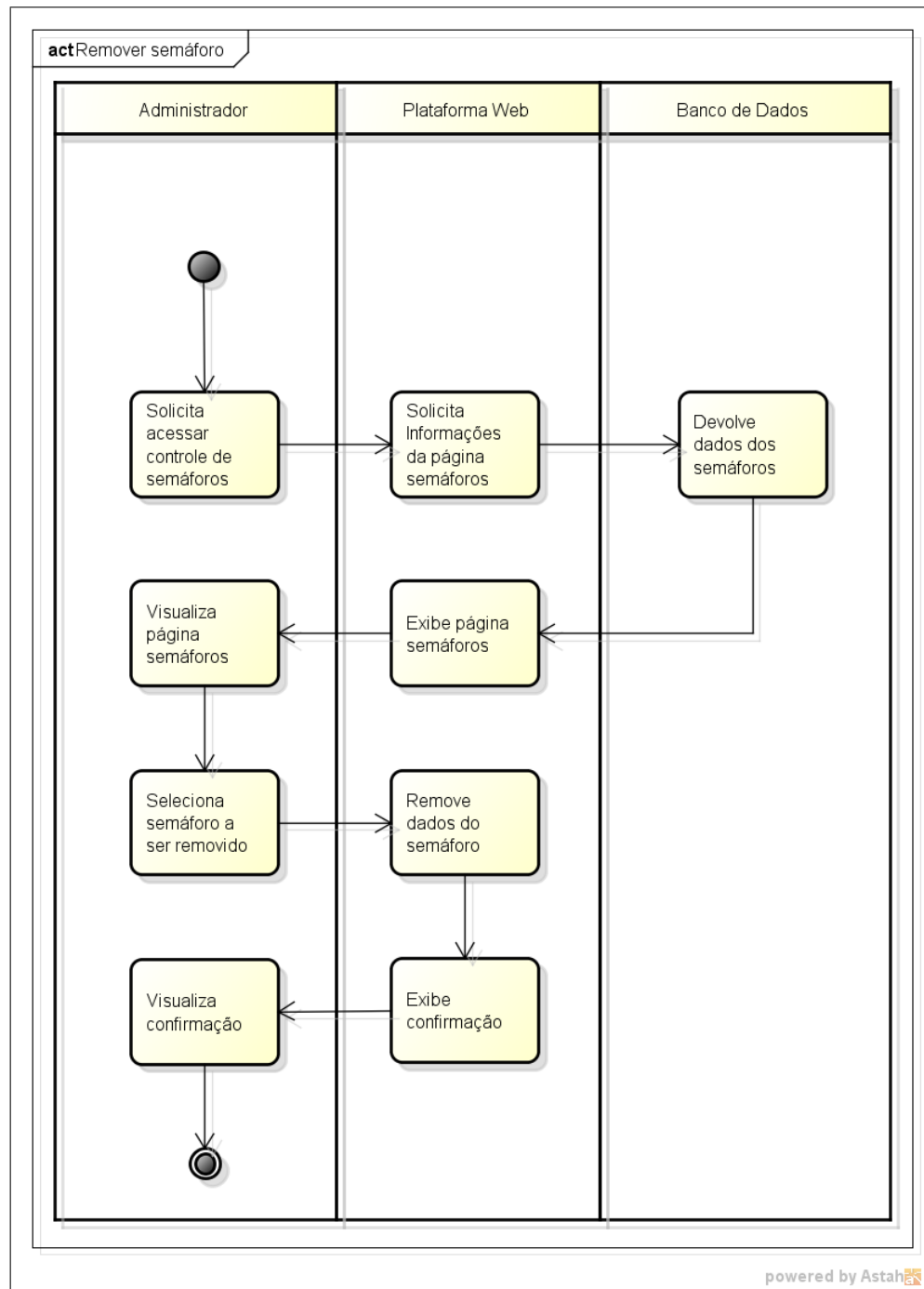


Figura 12 – Diagrama de atividades: remover semáforos

A consulta do estado dos semáforos (vermelho, verde e amarelo) é visualizada automaticamente quando a página de controle semafórico é acessada. Vale ressaltar que o estado do semáforo neste projeto é uma simulação ideal de acordo com os tempos definidos

pelos administradores. Uma vez inserido o semáforo, a sequência de estado inicia-se pela cor vermelha seguindo a cor verde e posteriormente a cor amarela de acordo com os tempos definidos para cada estado. A Figura 13 descreve esse processo.

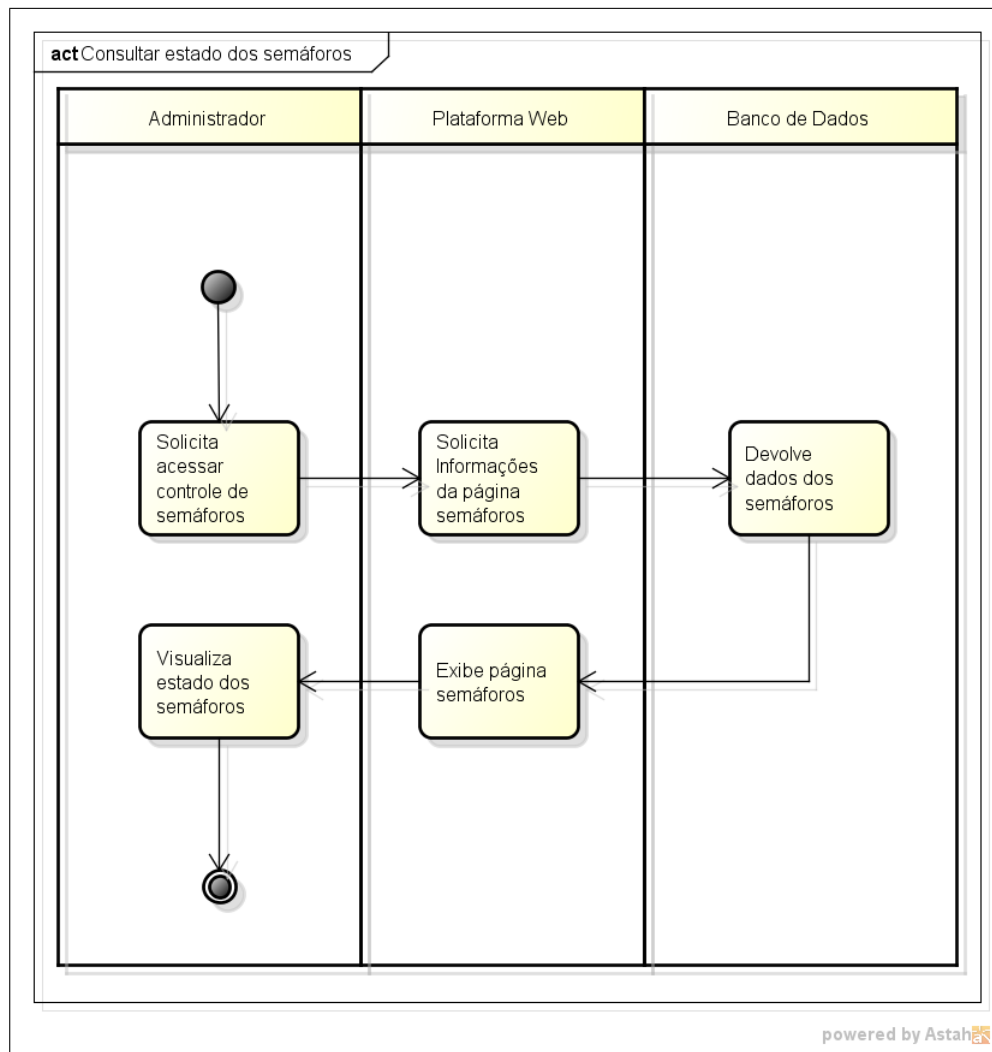


Figura 13 – Diagrama de atividades: consultar estado dos semáforos

3.2.1.3 Diagramas de classes

Além dos diagramas de casos de uso e atividades, é importante representar a estrutura e as relações das classes que servem de modelo para objetos. Assim, por meio do diagrama de classes, é possível verificar todas as classes que o sistema necessita. A Figura 14 ilustra essa representação.

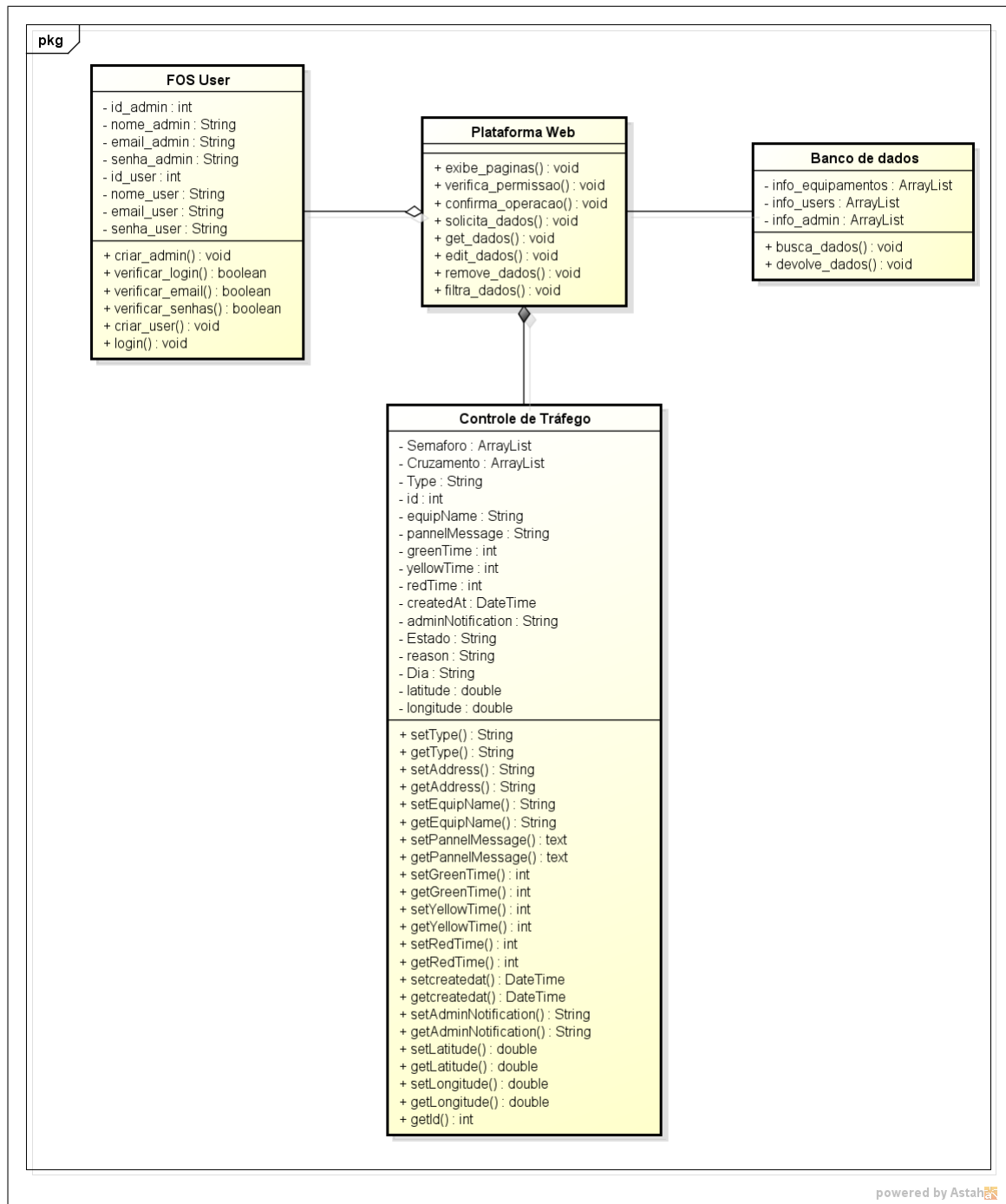


Figura 14 – Diagrama de Classes

Assim, de acordo com as funções e métodos que a plataforma web disponibiliza, resulta-se em 4 classes principais: FOS user, Plataforma Web, Controle de tráfego e Banco

de dados.

- FOS user - está associada diretamente ao login para acessar a plataforma web.
- Controle de tráfego - apresenta todos os métodos relacionado ao controle semafórico e outros equipamentos.
- Plataforma *web* - interface responsável por fazer a conexão com o banco de dados de acordo com a solicitação do administradores ou usuários.
- Banco de dados - apresenta todas as informações salvas pelos administradores ou usuários.

3.2.1.4 Diagramas de sequência

Diagramas de sequência foram utilizados para apresentar a interação entre processos/objetos mostrando as mensagens passadas entre eles. Dessa forma, ao longo do tempo é possível descrever como os grupos de objetos colaboram em determinado comportamento. Portanto, foram feitos principalmente diagramas de sequencia para o caso de uso em que o ator é o administrador: fazer *login*, inserir equipamento, editar equipamento, remover equipamento, inserir histórico e inserir notificações. Vale ressaltar que os equipamentos englobam semáforos, painéis eletrônicos, cameras e radares e os usuários comuns apresentam diagramas de sequencia iguais aos do administrador para *login*, visualizar tráfego, inserir notificações e visualizar histórico do tráfego.

A Figura 15 refere-se ao caso de uso *login*. Administradores e usuários comuns necessitam passar por essas etapas para poder acessar as páginas principais da plataforma *web*.

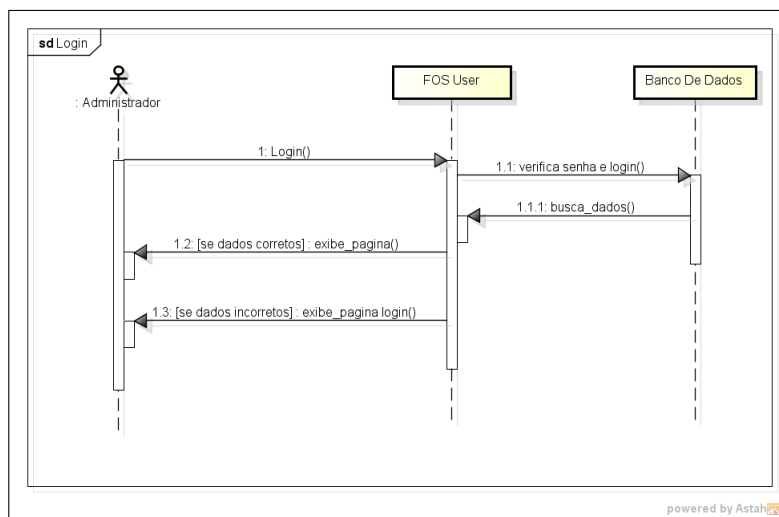


Figura 15 – Diagrama de sequência: *login*

As Figuras 16, 17, e 18 referem-se aos processos para registrar, editar e remover os equipamentos utilizados na plataforma *web*. Observa-se que os administradores tem controle sobre esses equipamentos e se desejar, por exemplo, modificar o tempo ou o estado dos semáforos é preciso primeiramente fazer o *login* e posteriormente editar o equipamento, no caso, os semáforos.

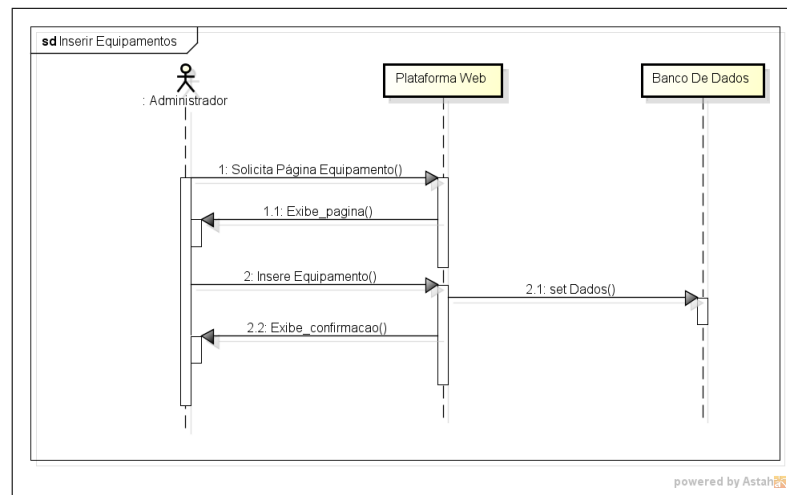


Figura 16 – Diagrama de sequência: inserir equipamentos

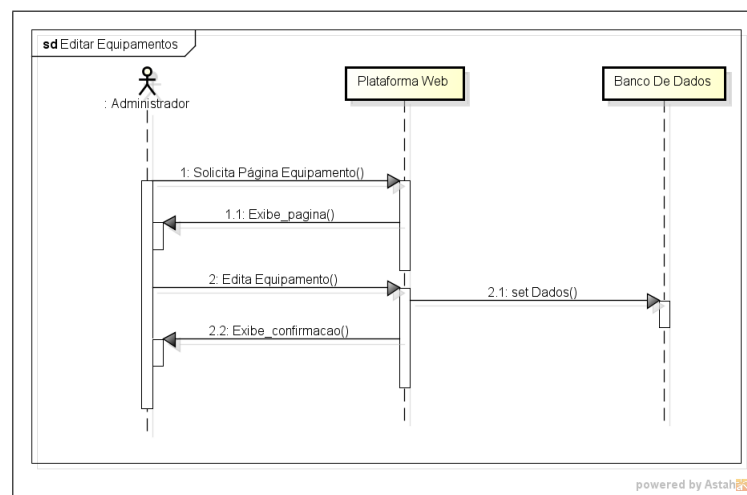


Figura 17 – Diagrama de sequência: editar equipamentos

Para inserir histórico do fluxo de trânsito em determinado endereço e ser visualizado por meio de gráficos é necessário seguir a sequência mostrado na Figura 19. Os administradores e usuários comuns podem acessar essa página porém somente os administradores podem inserir, editar e remover os históricos. Portanto, os usuários comuns podem somente consultar e visualizar os históricos por meio de gráficos.

Por fim, para a inserção de notificações sobre os problemas de tráfego a Figura 20 ilustra esse processo. Tanto os administradores quando os usuários comuns têm acesso a essa página para a inserção de notificações. Porém, cabe aos administradores a verificação,

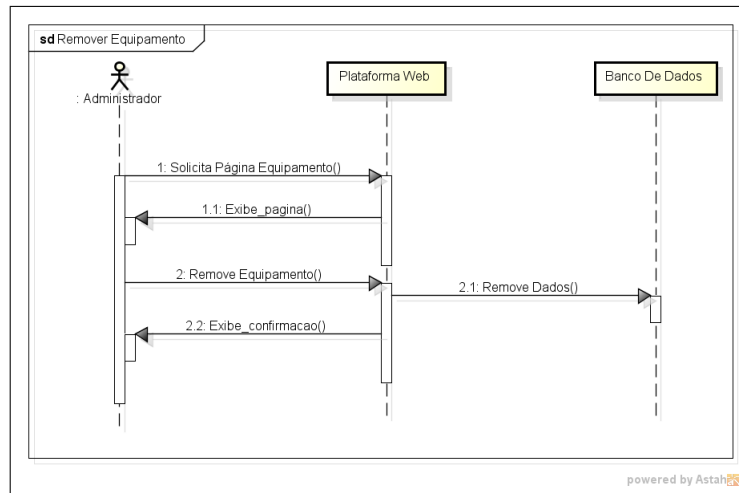


Figura 18 – Diagrama de sequência: remover equipamentos

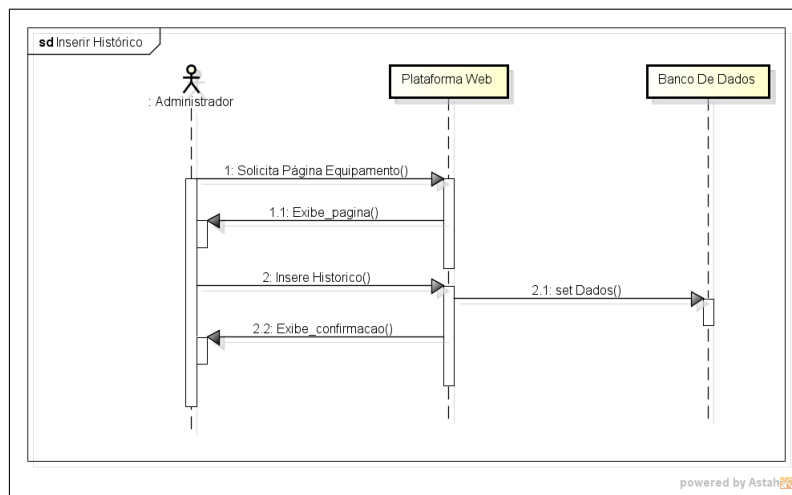


Figura 19 – Diagrama de sequência: inserir histórico

edição e remoção das notificações enviadas pelo usuário e o registro final na plataforma *web* para todos os demais usuários.

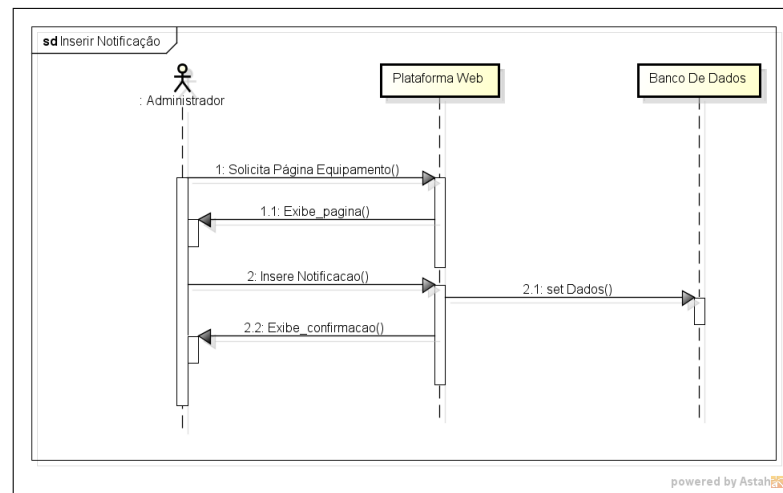


Figura 20 – Diagrama de sequência: inserir notificação

3.3 Ferramentas utilizadas

Baseado na ideia de implementar um controle e mapeamento de equipamentos do tráfego para um melhor gerenciamento do trânsito, o projeto é realizado somente em software. Assim, para o desenvolvimento do projeto utilizou-se bibliotecas e ferramentas que auxiliam na agilidade para a construção de um site. As principais ferramentas que serão descritas nos itens a seguir utilizadas foram: Apache como servidor, Symfony, FOS User e MySQL Workbench.

3.3.1 Servidor: Apache

Sendo uma plataforma *web*, o projeto engloba também a comunicação entre máquinas que utiliza a linguagem HTTP (*HyperText Transfer Protocol*). Dessa forma, o navegador comunica com o servidor através dessa linguagem por meio de diferentes métodos como *GET* e *POST* que são usados para enviar e receber, respectivamente, dados entre as páginas da aplicação. A comunicação tem basicamente dois passos, sendo o primeiro, um pedido feito pelo navegador e o segundo, a resposta feita pelo servidor. Esse processo pode ser visualizado na Figura 21.

O servidor utilizado neste projeto é o Apache (versão 2.2.8) e a linguagem utilizada para processar a requisição é o PHP (*HyperText Preprocessor*).

3.3.2 Symfony

Usando-se o *Sonata Project*, que é uma biblioteca que auxilia no desenvolvimento de plataformas *web*, existe uma ferramenta, que se chama *Symfony*, que permite construir aplicações mais robustas e rápidas. *Symfony* fornece uma alternativa para a comunicação PHP puro entre servidor e navegador. Ela baseia-se em duas classes, *Request* e *Response*,

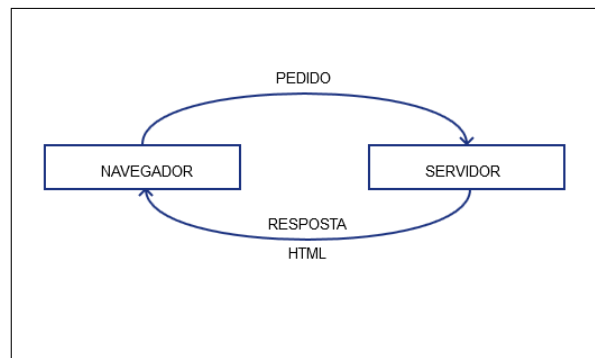
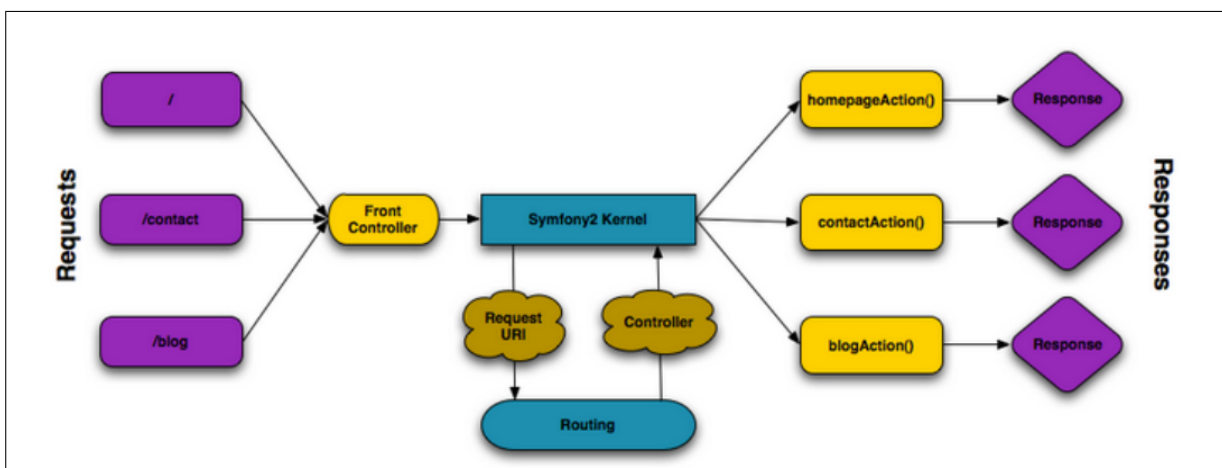


Figura 21 – Comunicação HTTP

que permitem uma interação entre navegador e servidor com a solicitação em HTTP. Além disso, Symfony é um sistema que controla de forma bem organizada a transição entre pedido e resposta que implementa um modelo *Model-View-Controller* (MVC). As solicitações recebidas são interpretadas pelo roteamento que mapeia as diferentes URLs para as diferentes funções em PHP. Cada função em PHP, conhecido como controlador, é responsável por utilizar as informações do pedido para criar e retornar um objeto da classe *Response*. É no controlador aonde se faz a programação (a interpretação da solicitação e também a criação de uma resposta). O controlador interage com o banco de dados e consegue manipular os dados apresentados e o enviar mensagens por *e-mail*.

A Figura 22 mostra como é a funcionalidade da ferramenta Symfony e é melhor compreendida nos itens descritos a seguir.

Figura 22 – Funcionamento da ferramenta *Symfony*

- Cada solicitação executa um arquivo do controlador principal (*Front Controller*).
- O sistema de roteamento determina qual função PHP deve ser executada.
- A função correta PHP é executada e assim tem-se a resposta.

Assim, foi possível desenvolver na plataforma *web* com vários componentes disponíveis no *Symfony* entre elas:

- *HttpFoundation* - contém as classes de solicitação e resposta, bem como outras classes de manipulação de sessões e upload de arquivos.
- *Routing* - sistema de roteamento rápido que permite mapear uma URI específica (por exemplo, `/createSemaforo`) com algumas informações sobre como esse pedido deve ser tratado (por exemplo, executar o método `createSemaforoAction()`);
- *Form* - um *framework* flexível para criação de formulários;
- *Validator* - um sistema para criar regras sobre os dados e posteriormente sua validação;
- *ClassLoader* - uma biblioteca de carregamento automático que permite que classes em PHP sejam usadas sem a necessidade de exigir manualmente os arquivos que contêm essas classes;
- *Templating* - ferramentas para renderização de modelos, manipulação de templates (um modelo é decorado com um *layout*);
- *Security* - biblioteca para manipulação de todos os tipos de segurança dentro de uma aplicação;
- *Translation* - um *framework* para a tradução de textos na aplicação;

3.3.3 FOS User

Além disso, existem outras ferramentas que foram utilizadas neste projeto como, por exemplo, a FOS User. Essa ferramenta também faz parte do Sonata Project e adiciona suporte para um sistema de usuários em banco de dados no *Symfony*. Ele proporciona uma estrutura flexível para gerenciamento de usuários que visa lidar com tarefas comuns, tais como o registro do usuário e de recuperação de senha. As suas principais características são:

- Suporte de registro com uma confirmação opcional por email.
- Redefinição de senha.

3.3.4 MySQL Workbench

O banco de dados utilizado é o MySQL que utiliza a linguagem SQL (*Structured Query Language*). Para a construção da base foi utilizado uma ferramenta presente no

MySQL Workbench, no qual através do diagrama de entidade relacionamento, cria a base do sistema por meio do esboço de tabelas. Assim, essa ferramenta gera os comandos na linguagem SQL para a construção física do bancos de dados da plataforma web. As tabelas utilizadas são representadas na Figura 23.

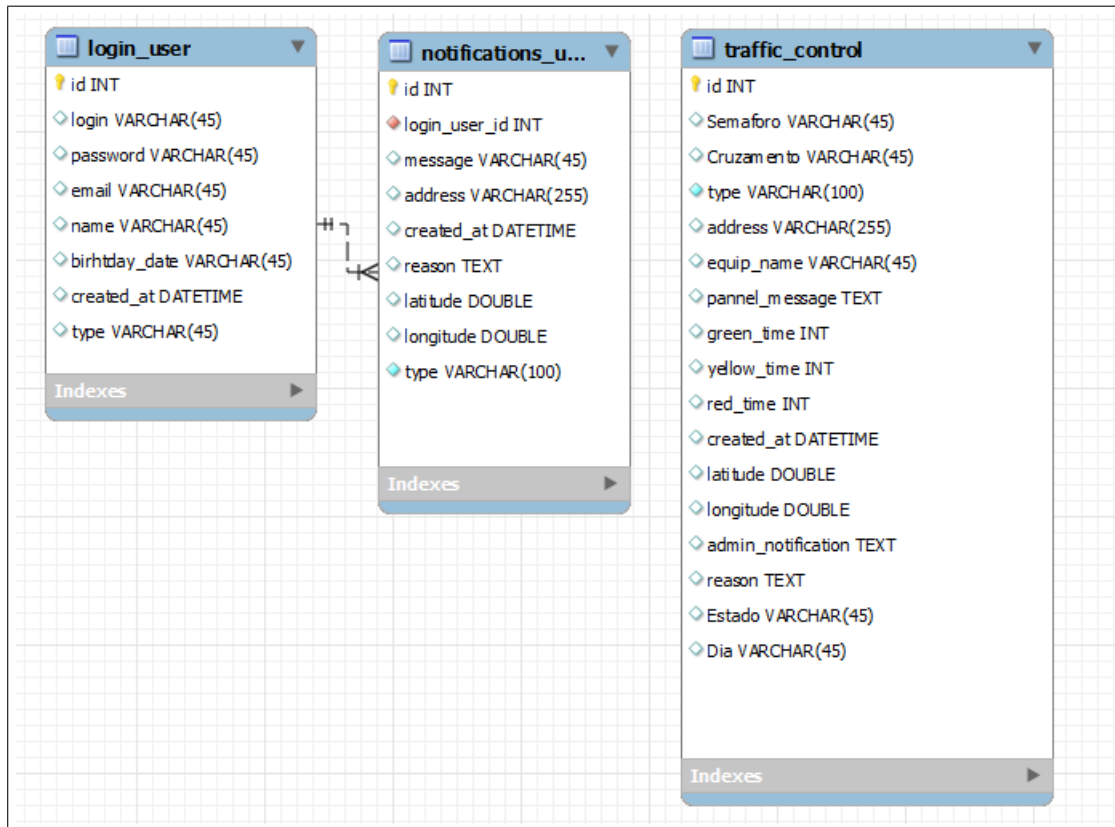


Figura 23 – Banco de dados em tabela, MySQL Workbench

4 Resultados

A utilização de diversas ferramentas como Symfony, FOS User e o programa MySQL Workbench foi fundamental para a agilidade do desenvolvimento desta plataforma *web* pois com elas existem propriedades que facilitam a construção rápida de um site. Além disso, com essas ferramentas é possível a comunicação entre as diversas linguagens em programação *web*.

O manuseio da API do Google Maps foi essencial para a possibilidade de gerenciamento e controle semafórico e mesmo o estudo do histórico do tráfego em determinado endereço com a ajuda da opção *Traffic Layer*. Assim, com a plataforma *web* é possível ter um controle geral dos equipamentos de tráfego e, além disso, possibilita uma comunicação com diversos usuários notificando os problemas de trânsito que ocorrem em tempo real.

O desenvolvimento do aplicativo é um ambiente teste para o gerenciamento do controle semafórico e de outros equipamentos, porém ainda não está integrado com todas as informações e dados reais de tráfego. O aplicativo suporta os recursos existentes no computador de quem o administra. Se esse aplicativo fosse integrado com o ambiente real da cidade de São Paulo, esse cenário mudaria, de acordo com o registro de informações, ou seja, o banco de dados.

4.1 *Layout*

O *layout* do aplicativo é bem organizado para uma melhor compreensão do administrador e usuário. Assim, a plataforma possui diversas páginas que são descritas nas próximas subseções. Como já citado anteriormente na seção 3.2, usou-se diversas linguagens para obter uma melhor interface com a plataforma *web*.

4.1.1 Página: *login*

A Figura 24 é a página de acesso ao controle e monitoração do trânsito. Se um usuário tentar acessar qualquer página sem o cadastro ou sem ter feito o *login*, a plataforma automaticamente será renderizada para esta página. Além disso, os usuários comuns podem se cadastrar por meio do preenchimento da seção usuário, email, senha e confirmação de senha. Vale ressaltar que o processo de *login* foi realizado com o pacote *FOS User* junto com o *Symfony*. A Figura 24 ilustra esta página.

Figura 24 – Página *login*

4.1.2 Página: visualização trânsito

Uma vez feito o *login*, a página a ser atualizada é correspondida para a visualização do trânsito da cidade de São Paulo em tempo real com a ajuda do Google Maps. Tanto os administradores quanto os usuários comuns tem acesso a essa visualização. Nesta página também é possível buscar determinado endereço para ver as condições de tráfego do local desejado.

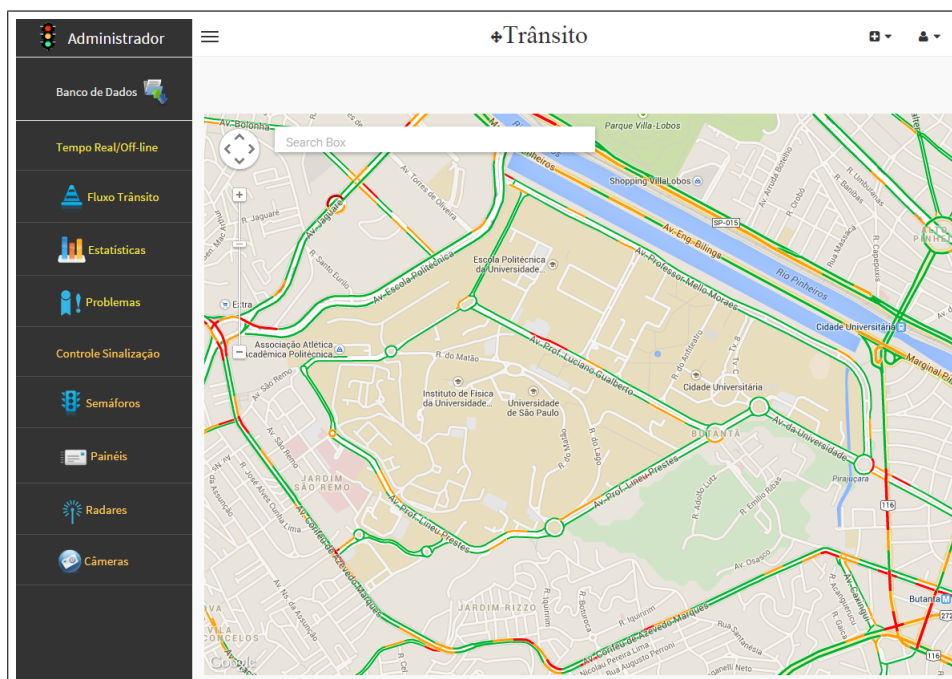


Figura 25 – Página visualizar tráfego

4.1.3 Página: estatísticas

No campo estatísticas, os administradores podem fazer um estudo do trânsito de acordo com as informações do Google Maps. Assim, é possível gerar e visualizar gráficos que indicam o histórico do trânsito em relação ao horário. A Figura 26 mostra o layout dessa página com o gráfico das condições de trânsito em determinado endereço.

O gráfico é resultado das condições de tráfego de acordo com a classificação de cores: verde, laranja, vermelho e marrom. Dessa forma, tem-se uma classificação de condições de trânsito sendo a cor verde representada por um fluxo de veículos sem problemas de lentidão e o marrom representado por trânsito lento com congestionamentos. É importante observar que o Google Maps interpreta as cores como a velocidade local dos veículos:

- Verde - 80 quilômetros por hora.
- Laranja - 40 à 80 quilômetros por hora.
- Vermelho - menos que 40 quilômetros por hora.
- Marrom - trânsito lento e congestionado.

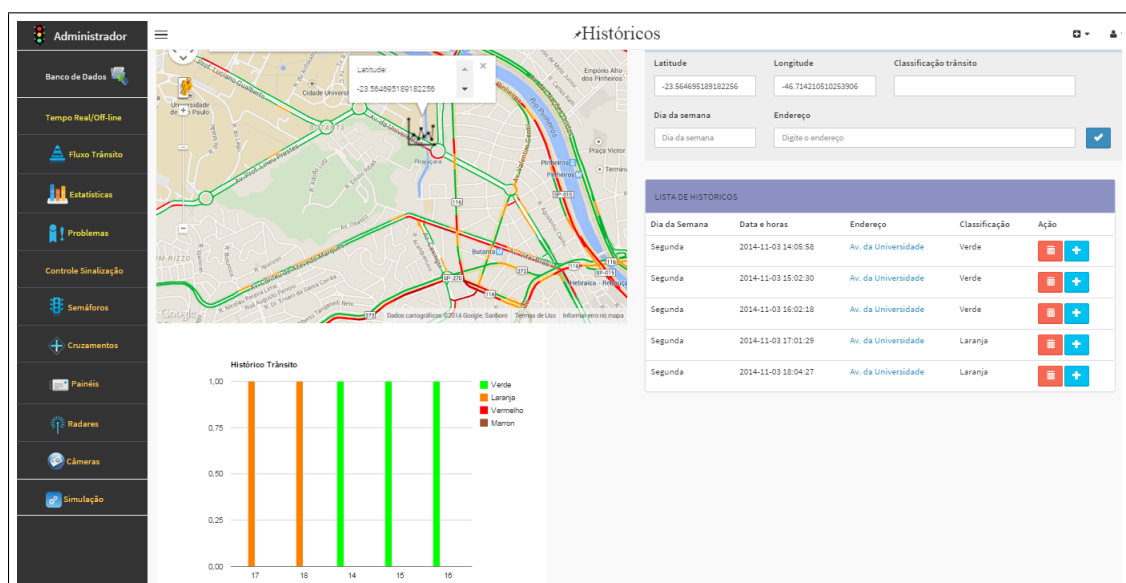


Figura 26 – Página estatísticas

4.1.4 Página: problemas

Já o campo problemas se refere aos incidentes de trânsito que ocorrem na cidade de São Paulo em tempo real. Assim, os administradores podem indicar a localidade dos problemas de tráfego em tempo real e notificar aos usuários na plataforma *web*. Dessa forma, os administradores recebem informações enviadas pelos usuários sobre o tráfego e

posteriormente, confirmam essas informações para todos os demais usuários. A Figura 27 é o *layout* desta funcionalidade.



Figura 27 – Página problemas

4.1.5 Página: semáforos

A página semáforos apresenta todas as ações que envolvem os semáforos como inserir, excluir e editar. Além disso, é possível visualizar o estado (vermelho, verde e amarelo) de cada semáforo na tabela que indica os semáforos inseridos. A Figura 28 mostra a interface dessa página com os administradores.

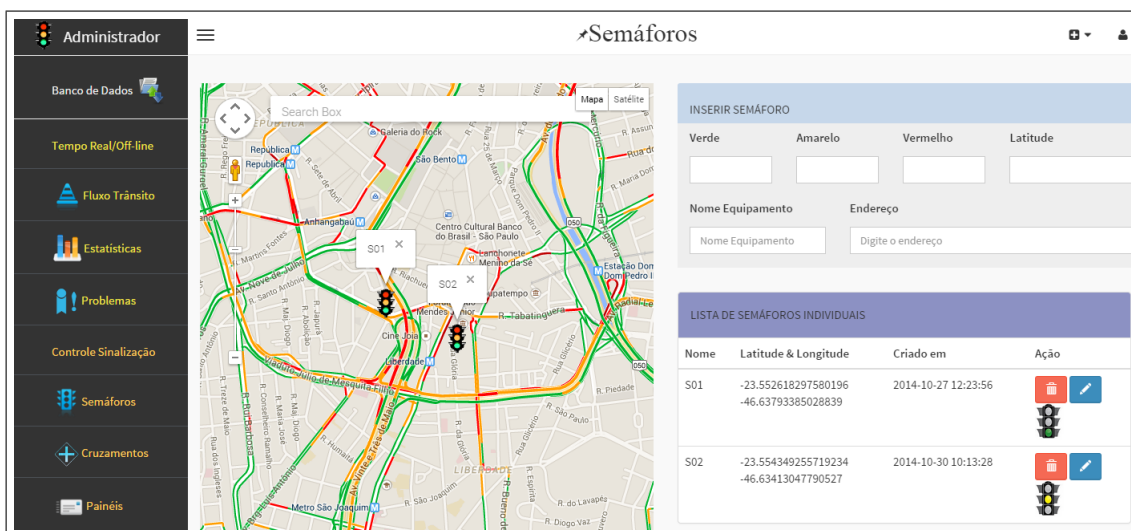


Figura 28 – Página semáforos

4.1.6 Página: cruzamentos

Por meio do ícone cruzamentos, os administradores podem inserir 4 semáforos em um cruzamento. A temporização desses semáforos é resultado do funcionamento real de um cruzamento com duas fases. A Figura 29 ilustra essa interface.

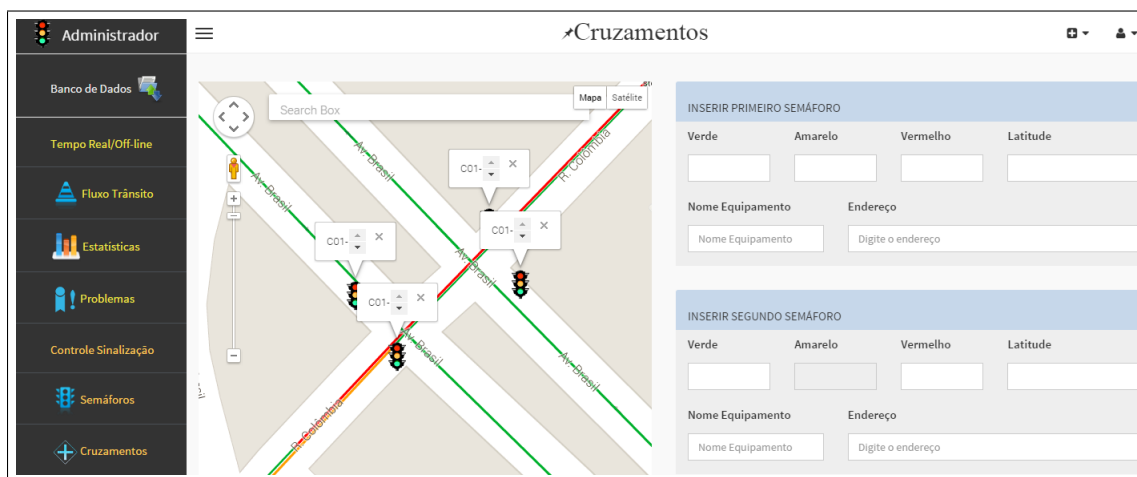


Figura 29 – Página cruzamentos

4.1.7 Página: radares

Acessando a página radares, é possível gerenciar e ter um melhor controle sobre os radares na cidade como por exemplo, suas localizações. As informações dos radares podem ser visualizadas em uma tabela do lado direito da página conforme a Figura 30.

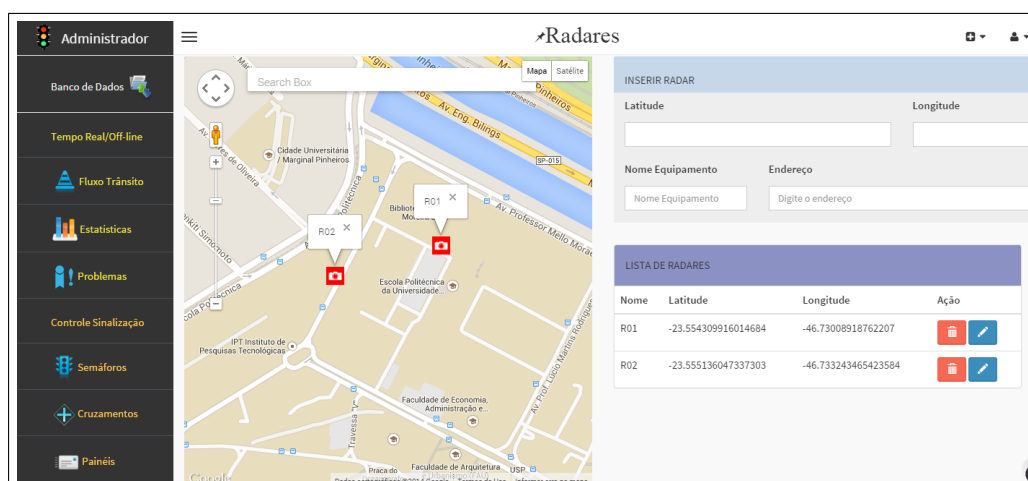


Figura 30 – Página radares

4.1.8 Página: câmeras

Como a página radares, este campo também permite um melhor gerenciamento sobre o equipamento, porém em relação às câmeras. A Figura 31 se refere ao *layout* desta

página.

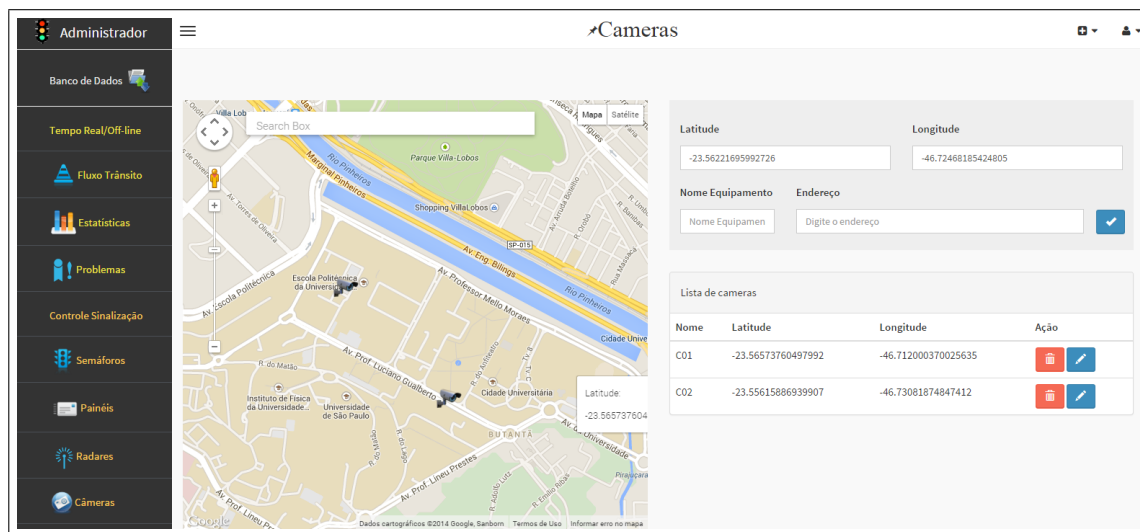


Figura 31 – Página câmeras

4.1.9 Página: banco de dados

Por fim, no campo banco de dados, os administradores tem controle sobre todos os dados que foram inseridos na plataforma. Assim, o administrador pode visualizar, editar, excluir todos esses equipamentos da plataforma de gerenciamento do trânsito. Nesta página, também pode-se filtrar e buscar os equipamentos desejados. A Figura 32 mostra o *layout* dessa página.

✓	Id	Tipo	Endereço	Nome	Mensagem	Tempo Verde	Tempo Amarelo	Tempo Vermelho	Notificação	Latitude	Longitude	Action
✓	46	Radar	Av. Eng. Bilings	R02						-23.554034537753	-46.722664833069	[Ver] [Editar] [Apagar]
✓	48	Problema	R. Alvarenga						Congestionamento	-23.560761470162	-46.712086200714	[Ver] [Editar] [Apagar]
✓	56	Historico	Av da Universidade							-23.56494104271	-46.713620424271	[Ver] [Editar] [Apagar]
✓	61	Semaforo	R. Alvarenga	S01		30	10	30		-23.565378660851	-46.711764335632	[Ver] [Editar] [Apagar]
✓	62	Painel Eletrônico	Marginal Pinheiros	P01	Transito Lento					-23.582159478883	-46.694340705872	[Ver] [Editar] [Apagar]
✓	63	Camera	Av da Universidade	C01						-23.564105138838	-46.714532375336	[Ver] [Editar] [Apagar]

Todos os elementos (6) [Apagar] [OK]

Download - 1 / 1 - 6 resultados - Por página 25

Filtros

Tipo

Endereço

Nome

Mensagem

Tempo Verde

Tempo Amarelo

Tempo Vermelho

Notificação

Id

[Filtrar] [Redefinir]

Figura 32 – Página banco de dados

5 Discussão

Inicialmente, o projeto tinha como objetivo somente o controle e gerenciamento semafórico. Com a agilidade das diversas ferramentas utilizadas para a implementação da plataforma *web*, foi possível incrementar outras funcionalidades para um melhor controle de tráfego. Com isso, implementou-se outras opções como o gerenciamento de painéis eletrônicos, câmeras, radares, histórico do trânsito e notificações de problemas em tempo real. Assim, tem-se uma visão mais geral do que se passa na cidade de São Paulo, permitindo que os usuários possam identificar os possíveis problemas de trânsito e, também, participar com novas notificações dos incidentes do tráfego por meio do *site*.

Utilizando o *traffic layer* do Google Maps permite ao administrador observar como está o trânsito em tempo real. Porém, é importante notar que os dados apresentados pelo Google Maps não são suficientes para interferir no controle de tempo do semáforo uma vez que os dados de tráfego demoram para ser atualizados. O que se tem é uma visão geral do trânsito. E com a ajuda dessa API pode-se adquirir novos dados, construir novas tabelas e ter um estudo melhor sobre como o trânsito se comporta nas principais vias de São Paulo.

A utilização de sistemas adaptativos em controle semafórico necessita de dados obtidos por detectores que são específicos de determinada localidade. Essa plataforma poderia ser integrada com esses dados medidos pelos sensores e dessa forma, ter um controle em tempo real da sinalização semafórica. A plataforma *web* desenvolvida poderia ser integrada com os semáforos já existentes.

6 Conclusões

É cada vez maior a pressão da população sobre as autoridades para que se desenvolvam políticas públicas que aprimorem a mobilidade urbana. O crescimento desenfreado do já expressivo número de veículos nos grandes centros urbanos torna indispensável o constante monitoramento e controle do tráfego por parte das autoridades. As centrais de controle visam otimizar o controle semafórico permitindo uma maior fluidez e segurança ao trânsito.

Apesar de já existirem centrais de controle de tráfego espalhadas pelo mundo e pelo Brasil, um aplicativo acessível via *web browser* se mostra uma excelente alternativa para gerenciar o trânsito nos centros urbanos. As vantagens desse tipo de sistema decorrem do seu baixo custo, flexibilidade e facilidade de implementação. A possibilidade de acesso via web ao sistema de controle trânsito permite que diferentes agentes de trânsito e usuários interajam de forma remota com o sistema ao mesmo tempo, eliminando a necessidade de um espaço físico para a central de controle.

A API do *Google Maps* se mostrou uma ótima ferramenta para servir de base para o aplicativo. Por meio desta ferramenta é possível adicionar e editar os instrumentos de controle semafórico diretamente em um mapa exibido na tela. Essa exibição no mapa torna o gerenciamento dos elementos de controle de trânsito muito simples e intuitivo, permitindo uma ampla visualização do sistema de controle e seus componentes no espaço.

Esse trabalho mostra que é possível desenvolver um aplicativo web para controle e monitoramento do trânsito usando-se apenas ferramentas gratuitas de livre acesso ao público. Um banco de dados foi criado e estruturado por meio do programa MySQL *Workbench*. A ferramenta Symfony foi utilizada para integrar o banco de dados, a API do *Google Maps* e outros elementos do *website* que foram construídos utilizando diferentes linguagens de programação. Os diagramas UML foram usados para permitir o entendimento e desenvolvimento das funções do *website*.

6.1 Trabalhos futuros

As sugestões de trabalhos futuros são as seguintes:

- Desenvolvimento de um aplicativo em plataforma Android que permita ao usuário permitir ao usuário e ao administrador utilizar as funções do aplicativo *web* em um *smartphone* ou *tablet*.

- Implementação de métodos de controle semaforico adaptativo a serem embutidos no aplicativo atual, permitindo um controle automatizado e otimizado do tráfego.
- Interligar os equipamentos (radares, semáforos, câmeras e painéis eletrônicos) com o software desenvolvido.
- Integrar a plataforma *web* com o aplicativo Waze que é um dos maiores aplicativos de tráfego e navegação.

7 Referências bibliográficas

AMBLER, S. W. The Elements of UML 2.0 Style – Agile Modeling. Estados Unidos: Cambridge University Press, 2005.

ANGELO, P. Relatório sobre a implantação do primeiro controlador com Protocolo UTMC tipo 2 na Cidade de São Paulo. São Paulo: Companhia de Engenharia de Tráfego, 2014. 33 p. (Nota Técnica da CET, 233).

BARNHART, C.; LAPORTE, G. Handbook in Operations Research and Management Science: Transportation. North-Holland, 2006. 796 p.

CHRISTIANINI, W.; HAGIWARA, S. S. Operação de trânsito: um desafio permanente. São Paulo: Companhia de Engenharia de Tráfego, 2008. 56 p. (Boletim Técnico da CET, 44).

CLOWES, D. J. Real-time wide area traffic control - the user's viewpoint. Traffic Engineering and Control. 1983.

DEPARTAMENTO NACIONAL DE TRÂNSITO. Manual de semáforos. 2 Ed. DENATRAN, 1984. 172 p.

FLOWLER, M. UML Essencial – Guia para a linguagem padrão de modelagem de objetos 3ª Ed. Brasil, 2005.

GORDON M. Traffic Devices: Historical Aspects Thereof. Washington D.C.: Institute of Traffic Engineers, 1971. 149 p.

GORDON, R. L.; TIGHE, W. P. E.; Traffic Control Systems Handbook. Washington D.C.: Federal Highway Administration, 2005. 367 p. (Relatório Técnico da FHWA).

HOMBURGER, W.S.; KELL, J.S.; PERKINS, D.D. Fundamentals of Traffic Engineering. 13 Ed. Berkeley: Institute of Transportation Studies, University of California, 1992.

JACOBSON, I. (1999). The Unified Modeling Language Reference Manual. 2 Ed. Addison-Wesley Professional, 2004. 721 p.

LARMAN, C. An Introduction to Object-Oriented Analysis and Design and Iterative Development, 2004.

MCSHANE, C. The Origins and Globalization of Traffic Control Signals. Journal of Urban History, 1999.

MELO, A.C. Desenvolvendo Aplicações com UML: do conceitual à implementação. Rio de Janeiro: Brasport, 2010.

PIAI, J. C . Estudo comparativo de técnicas de controle semafórico. 2009. 51 p. Dissertação (Mestrado) – Universidade Estadual de Londrina, Londrina, 2009.

ROESS, R. P.; PRASSAS, E. S.; MCSHANE, W. R. Traffic engineering. 3 Ed. Prentice Hall, 2004. 786 p.

SLINN, M.; MATTHEWS, P.; GUEST, P. Traffic engineering design. 2 Ed. Elsevier, 2005. 232 p.

SONATA PROJECT. Sonata Project reference, 1998. Disponível em:
<<http://sonata-project.org/>>.

SUTHERLAND, J.; SCHWABER, K. Scrum Guides, 2013. Disponível em:
<<http://www.ScrumGuides.org/>>.

SYMPONY, The Book reference, 1998. Disponível em:
<<http://symfony.com/pdf/>>.

VILANOVA, L. M. O controle de semáforos em tempo real: a experiência de São Paulo. São Paulo: Companhia de Engenharia de Tráfego, 2005. 70 p. (Boletim técnico da CET, 38).

Anexos

ANEXO A – UML

A.1 Introdução

De acordo com o objetivo do projeto, deve-se desenvolver um software em plataforma WEB e para isso, a UML (*Unified Modeling Language*) será utilizada para a modelagem do *software*, sendo necessário introduzir sua descrição. Em um *software* é necessário definir características como, requisitos, estrutura lógica, dinâmica de processos, comportamentos e mesmo necessidades físicas sobre o equipamento onde o sistema será instalado (FOWLER, 2005). Dessa forma, a UML será utilizada para descrever por essas características e elas devem ser estudadas antes do *software* ser implementado.

A UML é conhecida como um meio padrão utilizada por engenheiros no campo da informática e usa-se linguagens orientadas à objetos, também conhecida como programação orientada a objetos (OOP). Este tipo de linguagem se baseia na composição e interação entre diversas unidades de *software* chamadas objetos. Assim, tem-se como objetivo criar objetos que se relacionam e trocam mensagens entre si. É importante notar que nesse tipo de programação, definem-se um conjunto de classes que caracterizam os objetos do *software*. Além disso, cada classe é responsável pelo comportamento e os possíveis estados dos seus objetos, assim como a interação com os outros (LARMAN, 2004).

A UML é um resultado de vários trabalhos de *experts* após um processo de modificações desde dos anos 80 e, a partir de 2004, a versão UML-2.0 foi enunciada e mesmo assim, ainda passando por modificações e sendo utilizada globalmente. A UML pode ser dividida em diferentes aspectos para sua melhor compreensão tais como aspectos funcionais, aspectos estruturais e aspectos dinâmicos. Cada tipo de aspecto inclui diferentes diagramas de acordo com as características dos últimos. Inicialmente é descrito os aspectos a seguir e posteriormente, os diagramas serão descritos nas próximas seções (FLOWLER, 2005).

Aspectos Funcionais são representados através de diagramas de casos e sequências. Esse tipo de aspecto busca entender as interações entre diferentes atores/usuários e o sistema com o intuito de apresentar os cenários cronológicos e as interações típicas.

Aspectos Estruturais ou mesmo aspectos estáticos, são formados pelos diagramas de classes e diagramas de pacotes. Sendo que os primeiros favorecem a estrutura dos dados e identificam os objetos e componentes que constituem o programa como seus atributos, operações e métodos, e também suas ligações. Já os segundos têm o foco

em reagrupar as classes fortemente relacionadas em componentes mais autônomos possíveis. Dentro de cada pacote, há um diagrama de classes.

Aspectos Dinâmicos expressada pelo diagrama de estados. Esse aspecto tem como objetivo de descrever a evolução dinâmica dos objetos complexos do programa ao longo do seu ciclo de vida. Esse tipo de diagrama é uma espécie de fluxograma correspondente a uma versão simplificada do diagrama de estados. Ele permite representar as atividades que acontecem em paralelo uma das outras. Estabelecer um diagrama de atividades pode ajudar a desenvolver um diagrama de estado.

Além dos aspectos apresentados (Funcionais, Estruturais e Dinâmicos), existem também outros diagramas porém eles são mais ligados à implementação como diagramas de colaboração e diagramas de implementação. Primeiramente, os diagramas de colaboração estão mais associados com os aspectos funcionais e são considerados um pouco similar ao diagramas de sequências, porém menos ligado a sequência cronológica dos eventos. Já os diagramas de implementação está relacionada a uma implementação específica e indica sobre qual arquitetura serão desenvolvidos os diferentes processos que a aplicação realiza (JACOBSON, 1999). No entanto, para esse projeto não serão considerados esses dois aspectos apresentados por último.

Apresentado os aspectos, é importante detalhar cada tipo de diagrama. Assim as próximas seções tem como objetivo descrever os diagramas de caso de uso, atividades, classes e sequências os quais serão fundamentais para atingir o objetivo do projeto.

A.2 Caso de Uso

Um caso de uso UC modela a interação entre o sistema de informação a ser desenvolvido e o usuário que interage com o primeiro. Além disso, esse tipo de diagrama descreve uma sequência de ações realizadas pelo sistema que produz um resultado observável pelo usuário. Em general há dois tipos de descrição de caso de usos. A primeira é feita por meio da descrição textual de cada caso e a segunda é representada por um diagrama de casos que constitui uma síntese de conjunto de casos (FLOWLER, 2005).

Não existe uma regra precisa para dizer como deve ser a descrição textual do caso de uso. Geralmente cada caso de uso tem um nome, um pequeno resumo sobre o seu funcionamento, um contexto para o qual se aplica os usuários e posteriormente, uma descrição detalhada mostrando o funcionamento de todas interações, e em particular, se existe tratamento de casos especiais. Por fim, os efeitos ao desenrolar sobre o conjunto do sistema.

É interessante observar que existe ligações entre os casos mesmo sem passar pelo usuário. UML estabelece quatro tipos de relacionamentos: relação *Comunicação*, *Inclusão*,

Extensão e Generalização (FOWLER, 2005).

Comunicação representa a interação do usuário com o caso de uso e além disso esse tipo de relação permite com que um usuário possa se relacionar com mais de um caso de uso.

Inclusão é empregada quando dois casos de usos tem em comum tem uma mesma funcionalidade e deseja-se a criação de um sub-caso, ou seja, um caso intermediário para fornecer as diferenças de usos.

Extensão, esquematicamente é uma extensão do caso quando um caso é similar a um outro porém um pouco mais específico. Portanto, esse conceito é usado mais para identificar casos mais específicos.

Generalização pode ocorrer entre casos de uso e entre usuários. Entre casos de usos, demonstra uma sequência de comportamentos sendo que os casos de uso herdeiros utilizam os mesmos comportamentos dos casos mais genéricos.

A.2.1 Como é um Caso de Uso

A Figura 33 é um exemplo de diagrama de caso de uso. Esse tipo de modelo inicia-se a partir de um evento acionador do sistema que é gerado por um agente externo (usuário, atores). Com isso, acontece uma troca de eventos entre o agente, o sistema e mesmo outros agentes se o sistema apresentar. As interações entre os agentes e o sistema ocorrem até que aconteça a conclusão lógica do caso de uso.

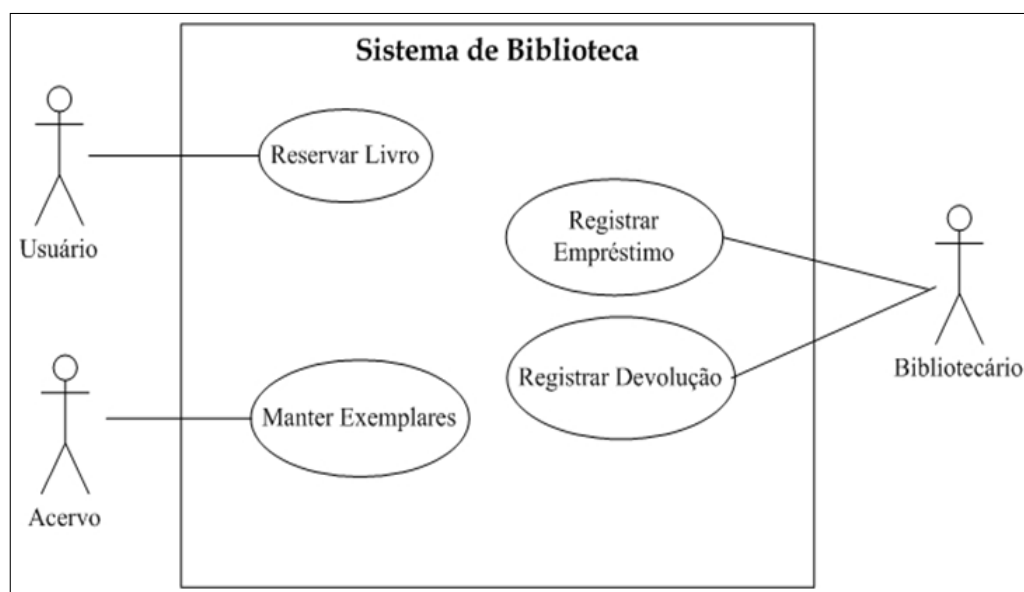


Figura 33 – Exemplo Caso de Uso

A.3 Diagrama de Atividades

Esse representa a dinâmica do sistema. Ele mostra a sequência de atividades de um sistema ou mesmo de uma operação. O diagrama de atividades representa um fluxo de controle que verifica o segmento de execução, transitando de uma atividade para outra dentro do sistema. Além disso, esse tipo de diagrama é centrado principalmente em atividades que alteram o estado de um sistema e está ligado a uma classe; a uma implementação de uma operação ou de uma utilização (AMBLER, 2005).

Um diagrama de atividades pode ser dividido em regiões. Estas regiões estão associadas a um objeto do modelo. Assim sendo, dentro de cada região, encontram-se as atividades relativas a estes objetos. As atividades por sua vez são conectadas por linhas que representam as transições e mostram as dependências entre elas (AMBLER, 2005).

O diagrama de atividades é um caso particular do diagrama de estados, no qual cada estado corresponde a uma atividade constituindo um elemento de tarefa global a realizar. O objetivo deste diagrama é de colocar em evidência as restrições na sequência e paralelismo que afeta na tarefa global (AMBLER, 2005).

A.3.1 Como é Diagrama de Atividades

A representação do diagrama é constituído por estados (inicial ou final, de ação e de atividade) e por transições que ocorrem quando a ação ou a atividade do estado se completa. Durante as transições pode existir guardas associadas à elas. Além disso, pode haver ponto de ramificação, ponto de união, barra de bifurcação e barra de união conforme representado na Figura 34.

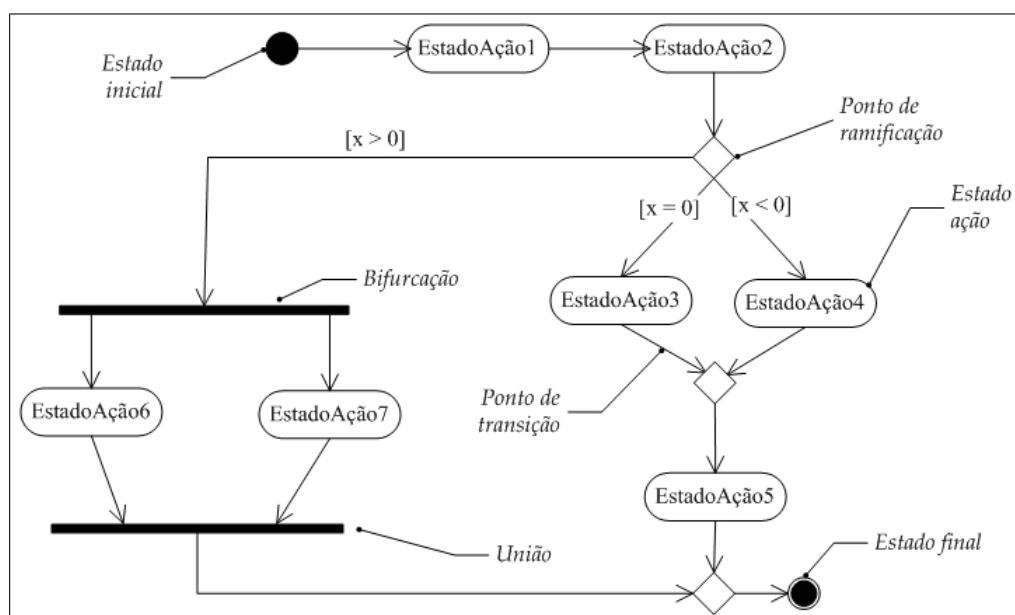


Figura 34 – Exemplo Diagrama de Atividades

O diagrama de atividades é usado para modelar o comportamento interno de uma classe, de um caso de uso ou de operação em uma forma de sucessões de ações. Também, é utilizado para representar uma sucessão de estados sincronizados e para modelar processo de negócios (*workflow*).

A.4 Diagrama de Classes

Este é o diagrama mais utilizado da UML, servindo de apoio para a maioria dos outros diagramas. Um diagrama de classes descreve o tipo de objetos ou dados do sistema e também as diferentes formas de relações estáticas que estabelecem entre eles. Dependendo do tipo de análise do sistema, o diagrama de classes muda drasticamente. O modelo de classes evolui ao longo das iterações de desenvolvimento e pode-se compreendê-lo em três principais etapas: análise, especificação e implementação (MELO, 2010).

Na fase de análise, as classes são consideradas relevantes pois o estudo é mais voltado para o que o sistema deve fazer. Além disso, não se considera as restrições inerentes à tecnologia a ser utilizada. Na fase de especificação, o modelo de análise é detalhado em classes com o objetivo de tornar um sistema possível. E assim, acrescentar detalhes às classes que foram identificadas na análise. Por fim, na fase de implementação, ocorre um detalhamento do modelo de classes e especificações e corresponde à implementação das classes em alguma linguagem de programação (MELO, 2010).

A.4.1 Como é um Diagrama de Classes

O diagrama de classes como mostrado na figura 35 é representando em classes e relacionamentos. As classes são caracterizadas em atributos, operações e responsabilidade. Já os relacionamentos, em associações, agregações e composições, generalizações e especializações (MELO, 2010).

Atributos é uma informação que a classe armazena. Operação é uma ação que um objeto da classe sabe realizar. Por fim, responsabilidade é o contrato ou a obrigação da classe (FOWLER, 2005).

As associações representam como se estabelece o relacionamento entre as classes e representa possíveis ligações entre os objetos das classes envolvidas. Agregações e composições representam o relacionamento todo-parte que indica que um dos objetos está contido no outro. E por último, generalizações e especializações representam um compartilhamento das similaridades das classes, preservando as suas especificidades.

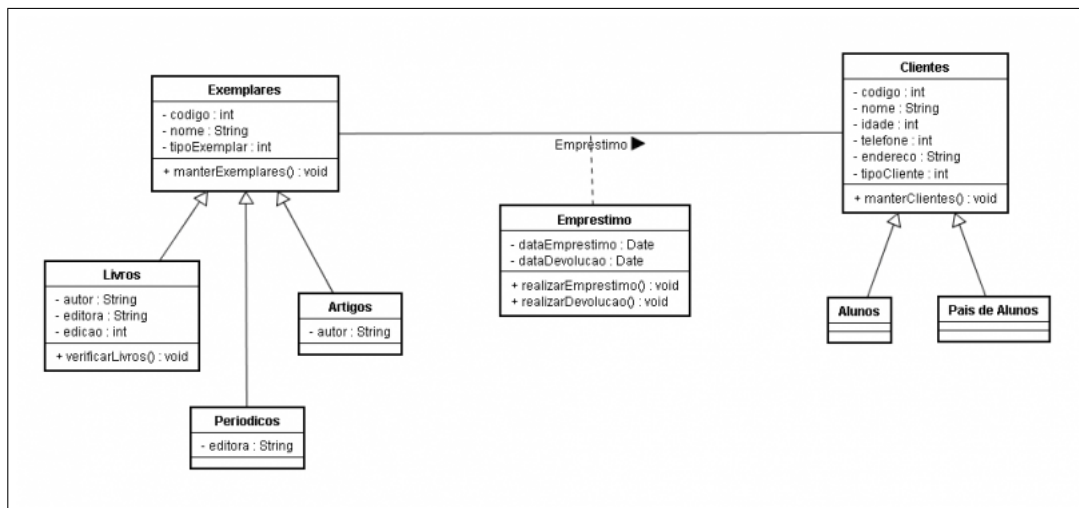


Figura 35 – Exemplo Diagrama de Classes

A.5 Diagrama de Sequência

Os diagramas de sequência enfatizam a troca de mensagens (desencadeando eventos) entre atores e objetos (ou entre objetos e objetos de forma temporal, a evolução do tempo de leitura de cima para baixo). Cada coluna corresponde a um objeto (descrito no diagrama de classes), ou, opcionalmente, um usuário, introduzido no diagrama de casos de uso. A linha de vida representa a duração do tempo com que a interação com objetos ocorrem.

Um diagrama de sequência é uma forma de capturar o comportamento de todos os objetos e usuários envolvidos em um caso de uso (LARMAN, 2004). Assim esse diagrama garante que todas as partes interessadas, classes, associações e as operações foram bem identificados nos diagramas de casos e de classes. Ele também é útil para a codificação de um algoritmo ou a criação de um controlador.

A.5.1 Como é um diagrama de Sequência

A Figura 36 representa um diagrama de sequência. De acordo com a definição anterior, os diagramas de sequência auxilia o projetista a se preocupar principalmente com o ordenamento temporal das ações. Dessa forma eles são construídos de acordo com convenções:

- Linhas verticais: representam objetos
- Setas horizontais: mensagens entre objetos
- Rótulos das setas: nomes de operações
- Posição vertical: ordem das mensagens
- Notações: utilizado para esclarecer o diagrama

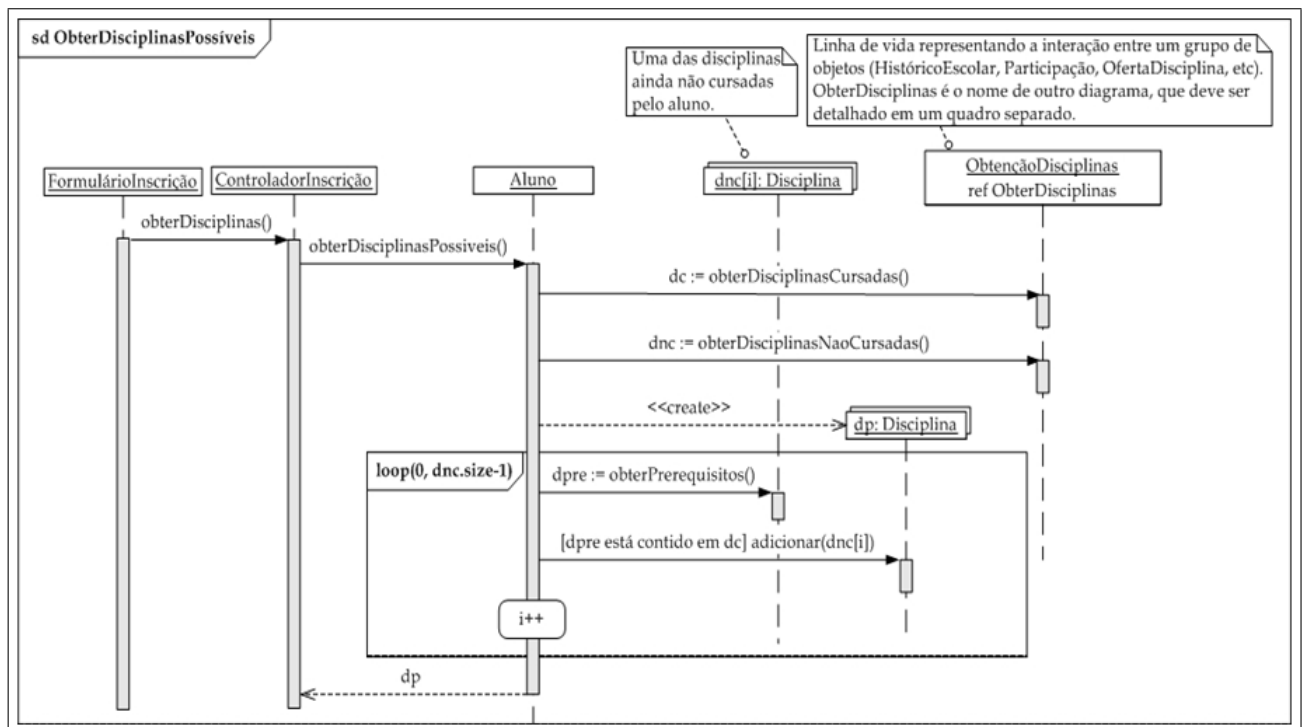


Figura 36 – Exemplo Diagrama de Sequência