

DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Sistema Computacional para Suporte à Operação Controlada do Transporte Coletivo Urbano

*Relatório submetido à Universidade Federal de Santa Catarina
como pré-requisito para a aprovação na disciplina
DAS 5511: Projeto de Fim de Curso*

Sergio Mauricio Prolo Santos Junior

Florianópolis, 7 de agosto de 2017

Sistema Computacional para Suporte à Operação Controlada do Transporte Coletivo Urbano

Sergio Mauricio Prolo Santos Junior

Esta monografia foi julgada no contexto da disciplina **DAS 5511: Projeto de Fim de Curso** e aprovada na sua forma final pelo **Curso de Engenharia de Controle e Automação**.

Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas
Programa de Graduação em Engenharia de Controle e Automação

Prof. Dr. Werner Kraus Junior

Banca Examinadora

Prof. Dr. Hector Bessa Silveira
Presidente da Banca

Prof. Dr. Carlos Barros Montez
Avaliador

Me. Lucas Zimmermann
Orientador no Laboratório

Prof. Dr. Werner Kraus Junior
Orientador no Curso

Sr. Alex Junior Gonçalves de Andrade
Debatedor

Sr. Rong Jiarui
Debatedor

Agradecimentos

Dedico esse relatório e meu trabalho aos meus avós, e minha querida família, devido seu incrível suporte durante todos os anos que estive em seus braços.

Agradeço aos meus dois mentores em trabalho, Lucas Zimmermann e Werner Kraus, por me apoiarem e indicarem o caminho de luz perante a ignorância e dificuldade, mesmo com o pouco tempo que lhes restava. Agradeço à Carolina Francener, por indicar estudos indispensáveis usados durante o projeto.

Também agradeço a esta universidade, seu corpo docente, direção e administração que deram a oportunidade à janela que hoje vejo em meu horizonte.

*“Medicina, lei, negócios e engenharia
são ocupações nobres para manter a
vida. Mas poesia, beleza, romance
e amor são razões para ficar vivo.”*

(Robin Williams - Sociedade dos Poetas Mortos)

Resumo

Este projeto visa a implementação de um sistema computacional de suporte para linhas de transporte público controladas. Busca-se orientar motoristas de ônibus no sentido da regularização dos intervalos entre veículos da mesma linha, através de um controle de velocidade. Prevê-se aspectos de cálculo das orientações de controle aos motoristas; do armazenamento, organização e divulgação de informações; e da construção de um ambiente para simular as condições de produção e consumo dos dados. A arquitetura do sistema computacional proposta é implementada sobre a simulação e resultados em relação ao intervalo entre veículos são observados. Esses resultados são comparados com os obtidos usando controle por retenção, buscando comprovar sua equivalência. A hipótese de que a influência semafórica é prejudicial em demasia ao controle de velocidade também é testada. É apresentado o desenvolvimento da interface usada para indicar recomendações de conduta ao motorista do veículo, uma aplicação para dispositivos móveis. Conclui-se que o sistema é efetivo, não ambíguo e apresenta características desejadas à operação. O controle de velocidade é efetivo somente se a influência semafórica for reduzida. Estudos futuros devem ser realizados para a implementação de um sistema de prioridade semafórica ótima.

Palavras-chave: Automação. Transporte público. Controle por velocidade. Apoio computacional.

Abstract

This project aims the implementation of a suport computational system for controled public transit lines. It seeks to orient bus drivers in order to regularize the headways between vehicles in the same line, through a velocity control. It is expected aspects of the calculation of drivers's control guidelines; the information's storage, organization and disclosure; and the construction of a ambient to simulate the conditions of data's production and consumption. The proposed computational system's arquitecture is implemented on the simulation and results in regard of the headways between vehicles are observed. Theses results are compared with the ones obtained using retention control, seeking to prove theis equivalence. The hypothesis that the traffic light influence is too damaging to the velocity control is also tested. It's stated the development of the interface used to guide conduct recommendations to the vehicle's driver, one mobile application. It's concluded that the system in effective, unambiguous and shows desired characteristics to the operation. The velocity control is effective only if the traffic light influence is reduced. Future studies must be performed to the implementation of one optimal semaphore priority system.

Key-words: Automation. Public transit. Velocity control. Computational support.

Sumário

1	Introdução	1
1.1	Objetivo	2
1.2	Contextualização do Projeto	2
1.3	Cronograma do Projeto	3
1.4	Estrutura do Documento	3
2	Materiais e Métodos	4
2.1	Módulos e Etapas de Trabalho	4
2.1.1	Simulação	4
2.1.2	Controle	4
2.1.3	Servidores	5
2.2	Linguagem de Programação Python	5
2.3	Biblioteca Kivy	6
2.4	Ambiente de Simulação dos Métodos de Controle	6
2.5	Controle de Intervalos	8
2.6	SincroBUS	9
2.7	GTFS	9
2.8	Unified Modeling Language	11
3	Desenvolvimento de Aplicações	13
3.1	Módulo de Simulação	13
3.1.1	Modelagem de Cenário e Linha	14
3.1.2	Exportação de Dados	15
3.2	Módulo de Servidores	18
3.2.1	Adaptação e uso do Servidor SincroBUS e Atis-Web	19
3.2.2	Desenvolvimento do Aplicativo Móvel	23
3.2.3	Arquitetura Global do Sistema Computacional	29
4	Desenvolvimento Matemático do Controle Híbrido	31
4.1	Controle a Eventos	32
4.2	Controle Discreto	34

4.3	Relação entre Controles e Criação do Controle Híbrido	38
4.4	Quantização da Indicação de Velocidade Desejada	39
5	Resultados	41
5.1	Módulo de Controle	41
5.1.1	Cenário Tradicional	41
5.1.2	Prioridade Semafórica Total	44
5.2	Módulo de Servidores	50
5.2.1	Aplicativo Móvel	50
5.2.2	Sincronismo dos Servidores e Apresentações	53
6	Conclusão e Discussão	55
	Bibliografia	57

Lista de Figuras

1.1	Cronograma proposto. A divisão de etapas est...	3
2.1	O fenômeno de <i>bus bunching</i> visto em ônibus ...	7
2.2	Um ônibus i em sua estação de parada k no m...	9
2.3	Diagrama que representa a arquitetura do Sinc...	10
2.4	Diagrama relacional entre os arquivos de um f...	11
3.1	Diagrama que representa o sistema de modo ger...	13
3.2	Linha de ônibus modelada na simulação, consis...	15
3.3	Seção da malha viária modelada, na região pró...	16
3.4	Interface do programador do <i>software</i> Aimsun...	17
3.5	Estrutura interna da plataforma de controle c...	19
3.6	Mapa do itinerário construído, como apresenta...	22
3.7	Diagrama UML de Casos de Uso para o Aplic...	24
3.8	Diagrama UML de Estados para o Aplicativo.	25
3.9	Diagrama UML de Sequência para as operaçõ...	26
3.10	Diagrama UML de Classes representando as e...	28
3.11	Arquitetura final do sistema computacional pr...	29
4.1	Veículos O_1 e O_2 e suas velocidades ao longo...	32
4.2	Um veículo e sua velocidade ao longo do temp...	34
4.3	Um veículo e suas velocidades real (preto), mé...	37
4.4	Diagrama de Blocos da relação entre Controle...	38
4.5	Faixas de operação do Erro de Velocidade E_i ...	40
5.1	Valores médios de intervalos observados, em s...	43
5.2	Valores médios de intervalos observados, em s...	46
5.3	Resultados gerados em simulação da malha c...	48
5.4	Resultados gerados em simulação da malha c...	49
5.5	Representações, como vistas na interface do a...	50
5.6	Possíveis representações, como vistas na inter...	51
5.7	Interface gráfica do aplicativo móvel desenvol...	52
5.8	Três <i>softwares</i> e interfaces, em tempo real, d...	53

Lista de Tabelas

3.1	Dados, seus formatos padrão e seus signific...	16
3.2	URL's usadas pelo Cerebus, seus objetivos d...	20
3.3	URL's adicionadas ao Cerebus para implement...	21
4.1	Variáveis usadas no texto e seus significados.	31
5.1	Características dos grupos de observações r...	42
5.2	Resumo dos resultados obtidos, por grupo, e...	44
5.3	Características dos grupos de observações r...	45
5.4	Resumo dos resultados obtidos, por grupo, e...	45

Lista de Abreviações

API *Application Programming Interface*, ou Interface de Programação de Aplicações

AVL *Automatic Vehicle Location*, ou Localização de Veículos Automática

BRT *Bus Rapid Transit*, ou Transporte Rápido por Ônibus

CSV *Comma-Separated Values*, ou Valores Separados por Virgulas

GTFS *General Transit Feed Specification*, ou Especificação Geral de Dados de Transporte Público

JSON *JavaScript Object Notation*, ou Notação de Objetos JavaScript

URL *Uniform Resource Locator*, ou Localizador Uniforme de Recursos

Lista de Nomenclaturas

i Indexador, também identificador, de ônibus

k Indexador, também identificador, de estação

D_k Distância da origem até a estação k , em metros

$a_{i,k}$ Instante de chegada do ônibus i na estação k , em segundos

$\hat{a}_{i,k}$ Instante previsto de chegada no ônibus i na estação k , em segundos

$d_{i,k}$ Instante de partida do ônibus i na estação k , em segundos

$h_{i,k}$ Diferença de tempo observada na estação k entre o ônibus i e o a sua frente, em segundos

$r_{i,k}$ Tempo de retenção do ônibus i na estação k , em segundos

$t_{i,k}$ Tempo de viagem do ônibus i no trecho entre a estação k e a próxima, em segundos

$w_{i,k}$ Instante de embarque/desembarque completo do ônibus i na estação k , em segundos

1 Introdução

O controle operacional dos movimentos de ônibus em uma linha é necessário, uma vez que estes têm dificuldade de manter o plano de horários ou intervalo planejado [1]. Essa dificuldade ocorre devido a variações no ambiente de operação da linha, como na velocidade média dos ônibus e nos tempos de embarque e desembarque de usuários. Os resultados de tais variações são agravados devido ao processo naturalmente instável dos ônibus: dado um ônibus atrasado, esse encontrará mais usuários à frente, aumentando o seu atraso; e dado um ônibus adiantado, esse encontrará menos usuários a frente, se adiantando ainda mais.

Mesmo em sistemas do tipo *Bus Rapid Transit* (BRT), esse problema continua a aparecer [2]. Nesses, os ônibus operam em corredores exclusivos e contam com plataformas de pré-embarque, tornando mais previsíveis os tempos de embarque e desembarque. Não obstante, continuam sujeitos a instabilidade, derivada das velocidades médias diferentes de acordo com cada condutor e dos encontros com semáforos em estados diferentes no caminho.

Uma solução para o problema da instabilidade operacional dos sistemas de transporte público é a implantação de estratégias de controle em tempo real. Uma estratégia de controle comumente usada consiste em reter um ônibus por um determinado tempo numa estação quando este estiver adiantado em relação ao quadro de horários ou com um intervalo inferior ao planejado [3]. A retenção pode melhorar o cumprimento do plano, reduzindo o tempo de espera dos usuários e melhorando sua distribuição entre os ônibus.

Outra estratégia de controle, também encontrada na literatura [4], é o controle da velocidade operacional do veículo. Assim como no controle por retenção, o objetivo é reduzir a velocidade de um veículo adiantado em relação ao quadro de horários. Essa estratégia está sujeita a limitações intrínsecas, como os limites máximos e mínimos de velocidade da via, mas possui a preferência dos usuários em relação ao controle por retenção [5].

1.1 Objetivo

Este projeto visa a implementação de um sistema para orientar motoristas de ônibus no sentido da regularização dos intervalos entre veículos da mesma linha, através de um controle de velocidade. Prevê-se uma aplicação para *tablets* capaz de mostrar orientações ao motorista sobre a velocidade a ser executada ao longo do itinerário.

Para se obter uma aplicação como a descrita acima, os dados de campo devem ser tratados no sentido de permitir o cálculo das orientações de controle com base no estado de todos os veículos de uma determinada linha. Uma vez que há tratamento dos dados, pode-se ampliar a usabilidade destes por meio do armazenamento (para consultas de planejamento com base em dados históricos) e divulgação (para orientação a usuários), a serem realizados em escalas de tempo distintas.

A proposta de trabalho prevê, além dos aspectos de cálculo das orientações de controle aos motoristas, do armazenamento e da organização da informação, a construção de um ambiente para simular as condições de produção e consumo dos dados. Assim, elimina-se a dependência do fornecimento de dados de operação em tempo real, disponíveis apenas através de operadores nem sempre dispostos a divulgá-los.

1.2 Contextualização do Projeto

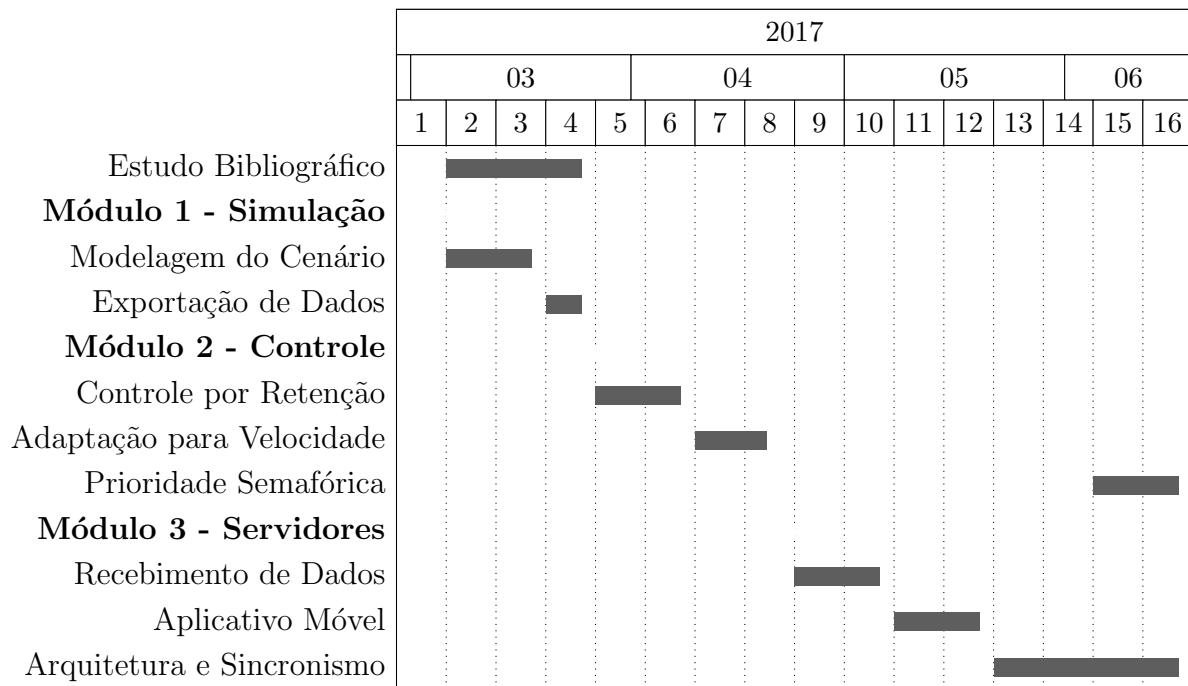
Os estudos e trabalhos executados durante o curso do PFC foram realizados no Laboratório de Tecnologias da Informação e Comunicação (LTIC), no Departamento de Automação e Sistemas (DAS) da Universidade Federal de Santa Catarina (UFSC). Estão vinculados com os estudos realizados sobre tráfego urbano inteligente e com o projeto SincroBUS [6].

O SincroBUS (Sistema de Informação e Controle em Tempo Real de Ônibus Urbanos) é um sistema de vínculo acadêmico dedicado ao desenvolvimento dos protótipos de *software*, aos serviços web necessários para a integração dos sistemas existentes e aos clientes web em geral.

1.3 Cronograma do Projeto

O cronograma foi criado usando o programa de agendamento e gerenciamento Gantt-Project [7], e é apresentado na Figura 1.1.

Figura 1.1: Cronograma proposto. A divisão de etapas está de acordo com a estrutura proposta na seção 2.1.



Fonte: Elaborada pelo autor

1.4 Estrutura do Documento

No segundo capítulo, os materiais, métodos e divisão de etapas de trabalho do trabalho são apresentados. O desenvolvimento de aplicações realizado no projeto, como a modelagem do cenário, exportação de dados, aplicativo móvel e arquitetura global do sistema, é apresentado no terceiro capítulo. O desenvolvimento matemático do controle híbrido criado no projeto é apresentado no capítulo quatro. Resultados obtidos são apresentados e discutidos no capítulo cinco. No capítulo seis é apresentada uma conclusão e realizada uma discussão geral do projeto.

2 Materiais e Métodos

Na atual seção, serão apresentados os materiais e métodos utilizados durante as atividades de estágio. Também é apresentada a divisão de trabalho adotada ao longo do trabalho.

2.1 Módulos e Etapas de Trabalho

Módulo foi a nomeação adotada para seções de trabalho a serem realizadas durante o período de projeto. Tais seções possuem vínculos entre si, mas foram trabalhadas separadamente ao longo do tempo e unidas quando possível. Isso se deu para facilitar a compreensão da extensão do trabalho e a criação de um cronograma adequado.

2.1.1 Simulação

O Módulo de Simulação trata da criação de um ambiente que simule a operação dos ônibus tal como ocorre na cidade. Os dados devem ser exportados como se fossem gerados por sistemas de *Automatic Vehicle Location* (AVL).

As etapas desse módulo são:

1. Construir o cenário de simulação e modelagem da linha de ônibus escolhida;
2. Implementar exportação de dados em formato padrão, estruturado como AVL.

2.1.2 Controle

O Módulo de Controle contém os algoritmos de maior complexidade matemática, responsáveis por tomar as decisões de controle. Parte deste módulo estava desenvolvida previamente, restando ao projetista adaptar as ações de controle que originalmente eram dadas em “tempo de retenção” para “velocidade média no trecho”.

As etapas desse módulo são:

1. Implementar o controle por retenção;
2. Adaptar os métodos para controle de velocidade;
3. Estudar a possibilidade de uso de, e possivelmente incluir, prioridade semafórica.

2.1.3 Servidores

O Módulo de Servidores é responsável pelo maior número de scripts do sistema computacional a ser desenvolvido. Consiste em aplicações para diferentes plataformas: aplicação para controle do simulador; Android; sistemas de interface web. Os sistemas web são responsáveis por apresentar informações em tempo real para os usuários. O comando de controle aos motoristas é passado na forma de recomendações de conduta. Esta informação é apresentada através de uma tela de Tablet Android ao motorista com ilustração intuitiva para que ele acelere ou desacelere o ônibus.

As etapas desse módulo são:

1. Receber dados do campo do tipo AVL (enviados pelo Módulo de Simulação);
2. Desenvolver a aplicação base para *tablets*;
3. Construir a arquitetura e sincronizar dados em sistemas para web.

2.2 Linguagem de Programação Python

Python [8] é uma linguagem de programação dinâmica, interpretada, de uso geral e alto-nível. A linguagem providencia módulos e estruturas com intenção de permitir a programação de códigos claros em larga ou pequena escala. Sua filosofia de projeto enfatiza a capacidade de compreensão e leitura, e sua sintaxe permite programadores expressarem conceitos complexos em poucas linhas de código.

Interpretores Python estão disponíveis para vários sistemas operacionais, permitindo códigos a funcionar em uma alta variedade de sistemas. A linguagem em questão possui uma biblioteca padrão de grande escala, geralmente citada como uma das grandes forças do Python.

Desde 2006, Python tem de modo consistente participado entre as cinco linguagens de programação mais populares de acordo com o Índice TIOBE [9]. Dentre as grandes organizações que usam Python, podemos incluir NASA, CERN e Google. Nas mídias sociais, o site de notícias Reddit é inteiramente escrito em Python.

As linguagens de programação que podem ser usadas em conjunto com o simulador usado no projeto são Python e C++. Python é ideal especialmente por sua característica

de alto nível e detalhada, seu processo de controle de memória usada e por fornecer uma estrutura facilmente interpretável. Tais características são ótimas em um programa que terão vários desenvolvedores, permitindo a evolução fácil e dinâmica das simulações de controle, sem a necessidade de um programador avançado durante o processo. Além, o autor do projeto possuía conhecimento prévio da língua, facilitando o trabalho. Portanto a linguagem Python é usada no projeto, ao invés de C++.

2.3 Biblioteca Kivy

Kivy [10] é uma biblioteca *open-source* de Python, com intenção de uso no desenvolvimento de aplicativos para portáteis e outras aplicações de toques múltiplos. Sendo uma biblioteca para aplicações multi-plataformas, é possível a criação de aplicativos para Android, iOS, Linux, OS X e Windows.

Sua *framework* contém elementos variados para a construção de uma aplicação, como: suporte de entrada de dados extensiva para mouse, teclado e eventos de toques; biblioteca gráfica usando apenas OpenGL ES 2; grande variedade de *Widgets*, elementos de aplicações, com suporte de toques múltiplos; e uma linguagem intermediária (Kv) [11] usada para facilitar a criação de *Widgets* personalizados.

O uso de tal biblioteca no PFC foi escolhido devido à facilidade da transição entre a aplicação da plataforma de controle utilizada e a criação da aplicação Android visada, ambas programadas em Python.

2.4 Ambiente de Simulação dos Métodos de Controle

O programa Aimsun (*Advanced Interactive Microscopic Simulator for Urban and Non-Urban Network*) [12] trata-se de um simulador microscópico interativo de trânsito urbano e não urbano. Assim sendo, é capaz de modelar o movimento dos ônibus enquanto se deslocam pelo itinerário pré-definido. São simulados os movimentos de velocidade constante; acelerações e desacelerações; possíveis mudanças de faixa em uma via; interações com outros veículos; e paradas em estações urbanas de transporte público.

Para o teste dos métodos de controle por retenção, uma tese de mestrado [2], realizada por um dos orientadores do PFC, propôs um cenário base implementado em ambiente de

simulação desenvolvido com o microssimulador de tráfego Aimsun. Esse cenário tem a capacidade de simular interações pontuais dos veículos, e permite o controle da simulação em tempo real através de sua Interface de Programação de Aplicações (API). A aplicação desenvolvida para essa API usa linguagem Python, e tem o objetivo de calcular o tempo de parada dos ônibus em cada estação em tempo real. A plataforma de controle foi reformulada para uma mais ampla abordagem e estrutura durante o estágio do autor do texto atual [13].

O tempo de parada nas estações pode ser pré-definido ou alterado dinamicamente durante a simulação através da API. A aplicação desenvolvida em [2] possui o objetivo principal de alterar o tempo de parada de cada ônibus. Para o trabalho atual, é prevista a adequação da plataforma para controle de velocidade e exportação de dados.

Figura 2.1: O fenômeno de *bus bunching* visto em ônibus de uma linha em Londres.



Fonte: Artigo on-line¹ sobre *bus bunching*

¹Disponível em <<http://thirtyseventech.com/2017/03/22/busbunching/>>. Acesso em Junho de 2017

2.5 Controle de Intervalos

A instabilidade natural do intervalo entre ônibus é elemento crucial na operação de uma linha de transporte público coletivo. Ela ocorre devido a variações na malha viária, no ambiente de operação da linha, e ao processo naturalmente instável da relação entre atraso e demanda em estações.

Essa característica pode, inclusive, causar o problema de comboiamento de ônibus, ou do inglês *bus bunching*, onde dois ou mais ônibus de uma mesma linha se encontram no mesmo ponto no mesmo instante, mesmo programados para um intervalo significativo de tempo. O resultados de uma linha de ônibus sem controle operacional é um serviço não confiável, tempos de espera em estações maiores e mais inconsistentes, bem como a lotação dos ônibus.

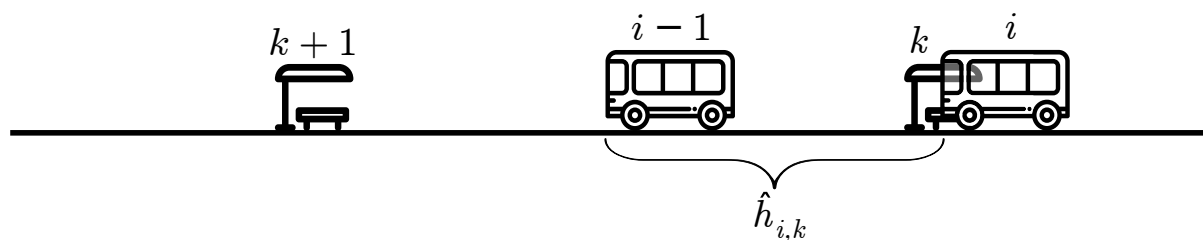
O Controle de Intervalos é o conjunto de relações entre os dados de campo e acionamentos que tem como objetivo regular o intervalo entre ônibus. Leva em consideração informações da malha viária, como distâncias entre estações, bem como de posições de ônibus da linha.

Originalmente, como é mais comum em aplicações práticas e estudos na literatura, o controle exerce sua influência através de retenções dos veículos em pontos de parada. A Figura 2.2 apresenta uma situação na qual é calculado controle por retenção. Nesta, o controle observa os ônibus i e $i - 1$ e o intervalos temporal $h_{i,k}$ entre esses. Subsequentemente, usa essa informação para encontrar qual deve ser o tempo de retenção $r_{i,k}$ a ser efetuado sobre o ônibus i . É importante notar que, por mais que a imagem indique uma representação espacial para o intervalo $h_{i,k}$, esse valor é calculado como a diferença de tempo observada em um mesmo ponto entre tais ônibus.

Contudo, o controle por retenção não está de acordo com os objetivos do projeto, uma vez que deseja-se indicar ao motorista recomendações de velocidade ao longo do tempo. Durante o curso do PFC, o autor desenvolveu um método de transformação do controle por retenção para um controle que concorda com os objetivos do projeto.

Este controle é chamado híbrido, uma vez que possui uma parcela orientada por eventos e outra discreta. Um evento ocorre toda vez que um ônibus chega em um estação de embarque e desembarque, onde uma velocidade média para o próximo trecho é calculada

Figura 2.2: Um ônibus i em sua estação de parada k no momento comum de cálculo de retenção.



Fonte: Artigo sobre estratégias de controle [2]

com base na retenção prevista para essa estação. Paralelamente, um controle discreto é acionado sempre que o ônibus não está em um ponto de embarque e desembarque, com intenção de garantir o seguimento da velocidade média calculada.

2.6 SincroBUS

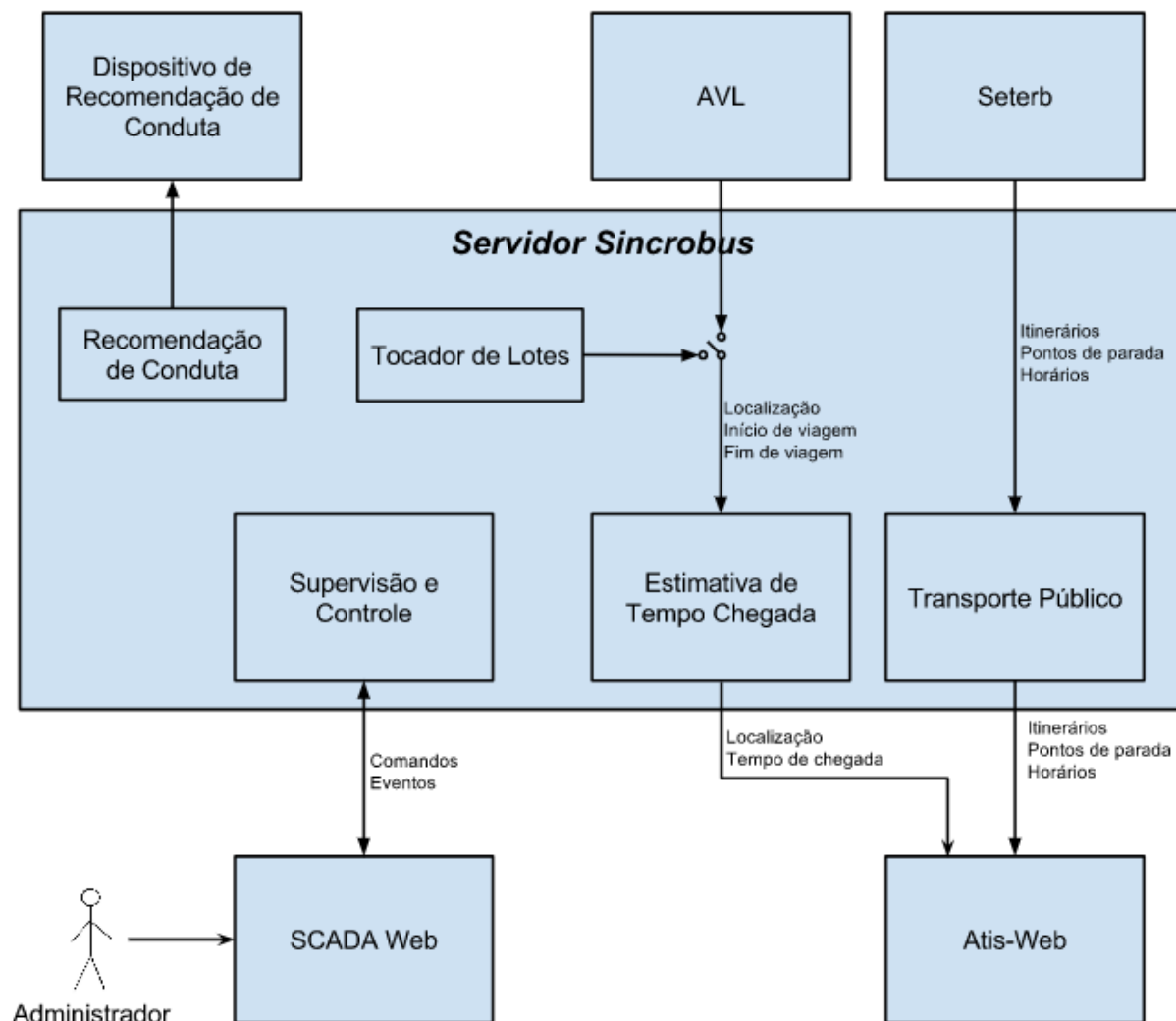
O SincroBUS [6] é um sistema dedicado ao desenvolvimento dos protótipos de software, aos serviços web necessários para a integração dos sistemas existentes e aos clientes web em geral. Durante o PFC, o aluno faz uso principal de dois módulos do sistema SincroBUS: o Servidor SincroBUS, ou Cerebus, que serve como mediador e agregador de informações e serviços; e o Atis-Web, um sistema Web voltado aos usuários de transporte público integrado.

O SincroBUS foi originalmente desenvolvido para uma aplicação sobre a malha viária de Blumenau, Santa Catarina. Durante o curso do PFC, o aluno também adaptou características do sistema para adequá-lo à malha modelada no Módulo de Simulação. A Figura 2.3 apresenta um diagrama que representa a arquitetura do SincroBUS. Importante notar que o módulo ‘Dispositivo de Recomendação de Conduta’ não foi criado pelo projeto SincroBUS; a criação de tal dispositivo é objetivo do projeto atual.

2.7 GTFS

A GTFS (Especificação Geral de Dados de Transporte Público) [14] define um formato comum para horários de transportes públicos e informações geográficas associadas. Os *feeds* GTFS permitem que as agências de transporte público publiquem suas informações

Figura 2.3: Diagrama que representa a arquitetura do SincroBUS. Durante o curso do PFC, o aluno fez uso do Servidor SincroBUS e módulo ATIS-Web. Notar que a especificação original do SincroBUS, fonte original dessa figura, era voltada a sua operação sobre a malha viária de Blumenau, Santa Catarina.

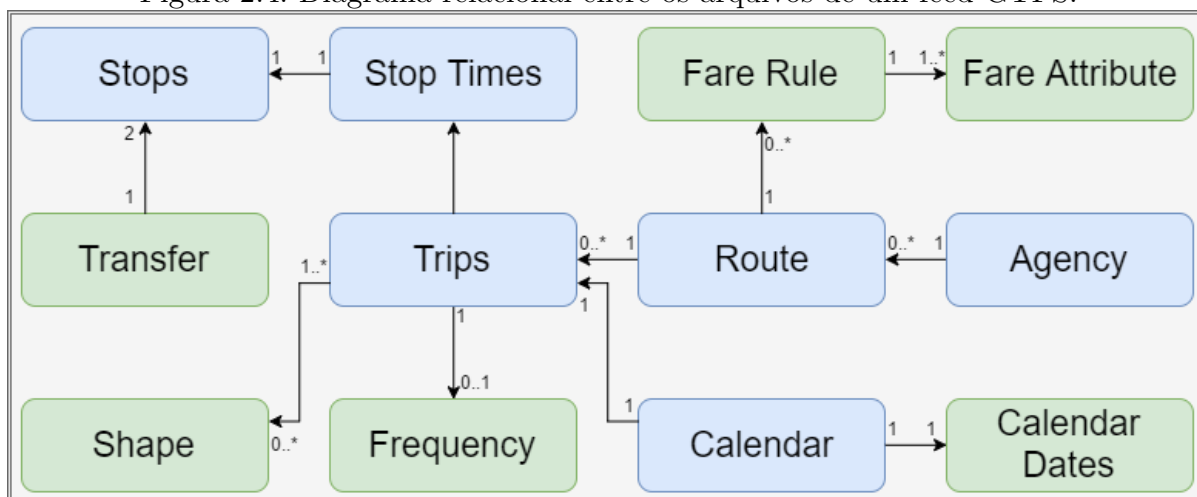


Fonte: Manual do SincroBUS [6]

e que os desenvolvedores criem aplicativos que consumam esses dados com interoperabilidade.

Um *feed* GTFS é composto de uma série de arquivos de texto, e o diagrama relacional entre tais arquivos é apresentado na Figura 2.4. Os arquivos representados na cor azul são obrigatórios para todo feed, enquanto os na cor verde são opcionais. Cada arquivo modela um aspecto específico das informações sobre o transporte público: paradas, trajetos, viagens e outros dados relativos ao plano de horários.

Figura 2.4: Diagrama relacional entre os arquivos de um feed GTFS.



Fonte: Elaborada pelo autor

Esse *feed* pode ser usado para oferecer recursos poderosos a planejadores de viagens, publicadores de tabelas e aplicativos gerenciais que usam informações sobre transporte público de alguma forma. Sua sintaxe é bem definida em formato padrão, ideal para o uso no PFC. Também, essa especificação é usada pelo Google Maps Transit e seus parceiros, característica que serviu como principal justificativa para o uso no PFC.

2.8 Unified Modeling Language

A UML (*Unified Modeling Language*) [15] é uma linguagem de modelagem padrão para a elaboração de projetos de *software*. A UML é descrita pela norma padrão ISO/IEC 19501:2004 [16]. Uma linguagem muito expressiva e abrangendo todas as visões necessárias ao desenvolvimento de sistemas, a UML permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados, especificando notações gráficas e sua semântica.

Essa linguagem não é uma metodologia de desenvolvimento, portanto não indica a ordem do que deve ser feito em passos subsequentes, mas auxilia o desenvolvimento e comunicação entre objetos à serem construídos. Os diagramas definidos pela UML usados no PFC são: Diagrama de Casos de Uso; Diagrama de Estados; Diagrama de Sequência; e Diagrama de Classes.

O primeiro é usado para auxiliar na compreensão de como um projeto de *software* é usado, seus casos de uso, quais são as operações sobre ele realizáveis e os atores responsáveis por realizar tais operações.

O Diagrama de Estados apresenta, de maneira sucinta, os estados em que o *software* pode se encontrar, e as consequências de todos os eventos que ocorrem em tais estados. Esse diagrama indica uma noção geral no funcionamento objetivo do programa.

O Diagrama de Sequência, por outro lado, indica temporalmente como cada operação ocorre no *software*, e o fluxo de informações que ocorrem entre cada setor de importância.

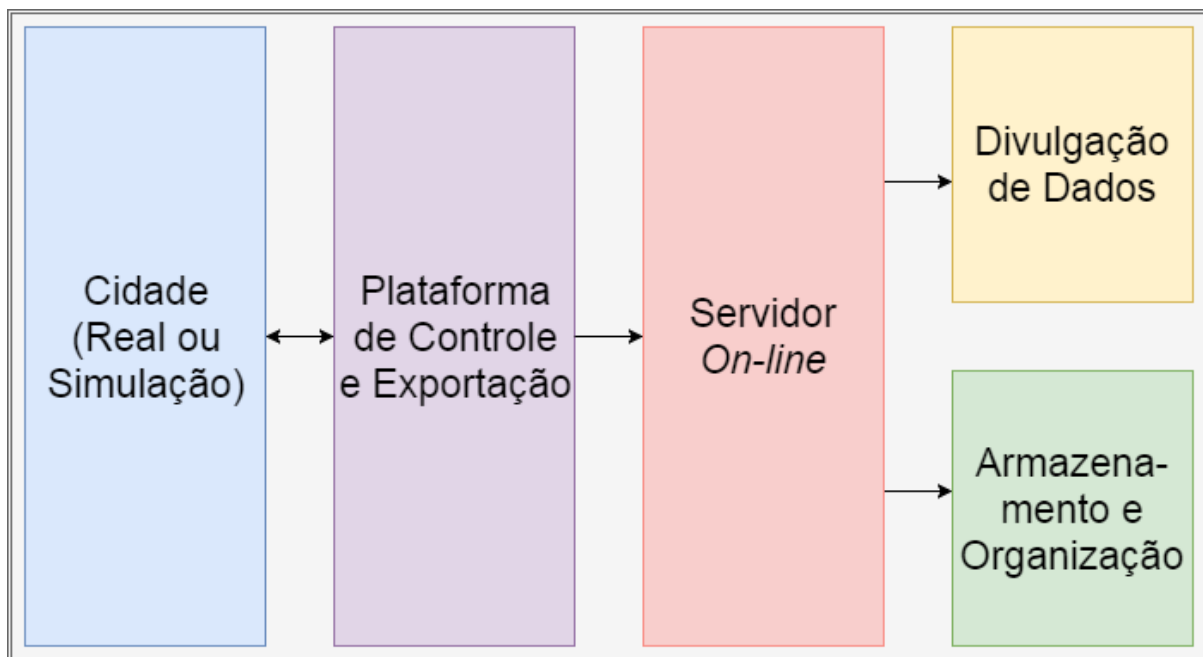
O Diagrama de Classes indica a representação da estrutura e relação das classes que modelam os objetos da programação realizada. A classe contém o nome do objeto, seus atributos (características) e métodos (comportamentos). Além disso, indica a associação entre classes, como multiplicidade ou hierarquia.

3 Desenvolvimento de Aplicações

Nessa seção será apresentado o desenvolvimento efetivo do trabalho realizado em relação aos Módulos de Simulação e Servidores. As aplicações desenvolvidas em tais módulos são compostas principalmente de *softwares* em Python, seja como programa voltado a API do microssimulador Aimsun, interface de comunicação web ou aplicativo para dispositivos móveis.

O objetivo das aplicações é gerar um sistema, de modo geral, como o apresentado na Figura 3.1. Esse diagrama serve de base para a construção final da arquitetura do sistema.

Figura 3.1: Diagrama que representa o sistema de modo geral. As setas representam o fluxo de informação. As informações da cidade são observadas pela plataforma de controle e enviadas ao servidor *on-line*, onde são organizadas e distribuídas.



Fonte: Elaborada pelo autor

3.1 Módulo de Simulação

O Módulo de Simulação é a seção de trabalho que comporta a criação de um ambiente que simule a operação dos ônibus tal como ocorre na cidade. Seus dados devem ser exportados de modo eficiente, definidos por um formato padrão.

3.1.1 Modelagem de Cenário e Linha

Com base no ambiente de simulação Aimsun, o autor desenvolveu um cenário de simulação com propósitos de gerar dados aos módulos seguintes e facilitar a visualização da operação da linha à ser modelada.

O cenário de simulação escolhido foi o anel viário do centro de Florianópolis, como o apresentado na Figura 3.2, incluindo as modificações propostas [17] para a implementação de corredor exclusivo para ônibus na região entre as estações 4 e 10. Pertencendo à região central de Florianópolis, o anel viário é parte da região metropolitana da cidade, e é uma das regiões com maior número de passageiros semanais [18]. O BRT proposto apresenta características desejadas para a operação de linhas controladas como a estudada no PFC, como maior previsibilidade de tempos de viagem e plataformas de pré-embarque.

A linha de ônibus escolhida para ser modelada é uma proposta de modificação da linha operacional 185 - UFSC - Semidireto, consistindo de 14 estações e 16km de extensão. As linhas de ônibus em Florianópolis não operam com base em intervalo de partida fixo, e sim em tabelas de horários de partida. Contudo, a linha em questão possui, em horários de pico de usuários, uma tabela horários com intervalos bastante homogêneos, semelhante à um intervalo fixo de partida. Portanto, essa linha é ideal para o estudo do controle de intervalos.

O aluno implementou a malha viária a partir de um arquivo pré-existente, contendo somente a proposta de corredor exclusivo para o anel viário de Florianópolis. Foi criado fluxo de tráfego misto na região entre as estações 5 e 13. A presença deste tipo de tráfego impõe perturbações propositais ao movimento dos ônibus. Também foi criada uma linha de ônibus fictícia que trafega entre as estações 6 e 12, também para impor perturbação ao movimento. Entre as estações 13 e 5 os ônibus trafegam exclusivamente.

Todos os semáforos, grupos semaforicos e controles de semáforos foram implementados de acordo com as estruturas reais. O corredor exclusivo ainda não foi criado, então seus grupos e controles foram criados de modo intuitivo. Nenhum estudo de campo foi realizado.

A Figura 3.3 apresenta uma seção da malha viária modelada na região próxima à UFSC, onde pode-se observar a duplicação e criação do corredor exclusivo propostos. A

Figura 3.2: Linha de ônibus modelada na simulação, consistindo de 14 estações e 16 km de extensão. Representa uma proposta de modificação da atual linha 185 - UFSC - Semi-direto.



Fonte: Google Earth Pro, editada pelo autor

Figura 3.4 ilustra a interface do Aimsun, onde o usuário da aplicação pode criar, modificar, deletar ou adicionar elementos da malha viária, bem como observar os resultados da simulação.

3.1.2 Exportação de Dados

Faz parte do Módulo de Simulação a criação de um método de exportação de dados sobre os ônibus em formato padrão, simulando um sistema AVL. A criação de tal método tem como objetivo fundamentar quais são as informações de importância dos veículos em um cenário como o modelado e armazená-las eficientemente. Isso permite acesso posterior adequado, facilitando estudos e modificações no sistema.

Figura 3.3: Seção da malha viária modelada, na região próxima à estação 6.



Fonte: Elaborada pelo autor

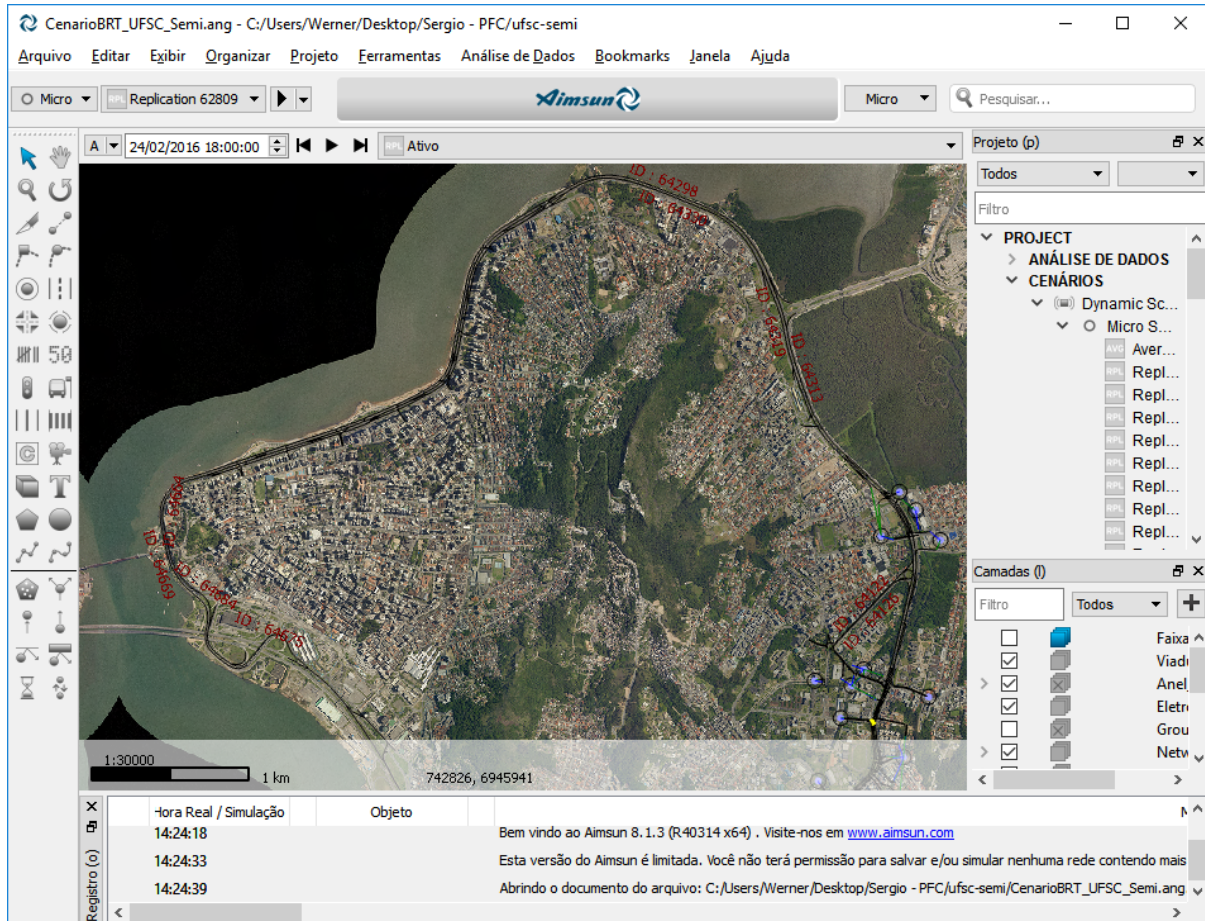
A plataforma de exportação de dados construída analisa o cenário modelado com referência aos dados da Tabela 3.1 e os armazena em seus formatos padrão usados no PFC. O sistema de exportação deve ser capaz de armazenar dados em memória, como um formato *feed* GTFS para uso posterior; ou ser enviá-los à um servidor *online*, como o SincroBUS.

Tabela 3.1: Dados, seus formatos padrão e seus significados no contexto do trabalho.

Dado	Formato Padrão	Significado no contexto	Exemplo
Posição	Latitude, Longitude	Local de interesse	'-27.4453, 45.2988'
Tempo	Padrão ISO 8601	Data e hora de um evento	'2017-05-11T13:31:27Z'
Ônibus	Valor Inteiro	Indexador de um ônibus	'3'
Estação	Valor Inteiro	Indexador de uma estação	'3'
Indicação	<i>String</i>	Recomendação de conduta	'MANTER_RITMO'

O microssimulador Aimsun utiliza um plano cartesiano para referenciamento da posição de elementos. Nele, todo elemento pertencente à simulação possui uma tupla de valores indicando a distância em metros, sob eixos perpendiculares, entre esse e um ponto de referência. O formato padrão para informações de posição, contudo, utiliza o georreferenciamento padrão com base de referencia no Meridiano de Greenwich e no Paralelo do Equador.

Figura 3.4: Interface do programador do *software* Aimsun. Nessa interface várias características da malha viária podem ser editadas e modificadas, bem como a simulação pode ser controlada e observada. O *software* também possui um terminal onde informações do programa em sua API podem ser visualizadas.



Fonte: *Print screen* da aplicação, sistema operacional Windows 10

Portanto, é necessária uma conversão dos dados, entre o plano cartesiano do simulador e o georreferenciamento padrão. Essa conversão foi realizada de acordo com as Equações de Haversine [19], formulas de importância devida em navegação e referenciamento global.

Durante a simulação, o *software* possui um valor interno de tempo. A simulação começa em um valor pré-definido pelo usuário, e ocorre durante um tempo específico. A simulação pode ser acelerada ou retardada, afim de se observar os eventos macro ou microscópicos com maior clareza, respectivamente. Esse valor interno de tempo é armazenado em segundos. Contudo, a exportação de formato padrão para tempo utilizado o Padrão ISO 8601 [20], onde os dados são indicados em ordem decrescentes de magnitude, ou seja, do maior (ano) para o menor (segundo).

Assim sendo, a plataforma de exportação armazena o tempo em formato padrão com sua parcela de data a partir da data real, com base na referência interna do computador do usuário. A parcela de tempo, por outro lado, pode ser armazenada a partir de dois princípios: transformar a representação interna da simulação em horas, minutos e segundos, para uma representação fiel à simulada; ou usar o valor de tempo real da simulação com base na referência interna do computador do usuário, para permitir armazenamento e divulgação da aceleração ou retardamento feita ao simular. A escolha de qual princípio é feita pelo usuário.

Em particular, uma recomendação de conduta dada a um motorista de ônibus é exportada em formato de *string*, uma lista ordenada de caracteres conhecidos. Tal *string* pode assumir um dentre quatro valores: ‘MANTER_RITMO’, ‘ACELERAR’, ‘DIMINUIR’ ou ‘SEM_DADOS’. A definição de tais *strings* foi realizada na criação do SincroBUS, e o projeto segue o modelo proposto. Qual recomendação é usada a cada instante de tempo para cada ônibus deve ser definido pelo sistema de controle.

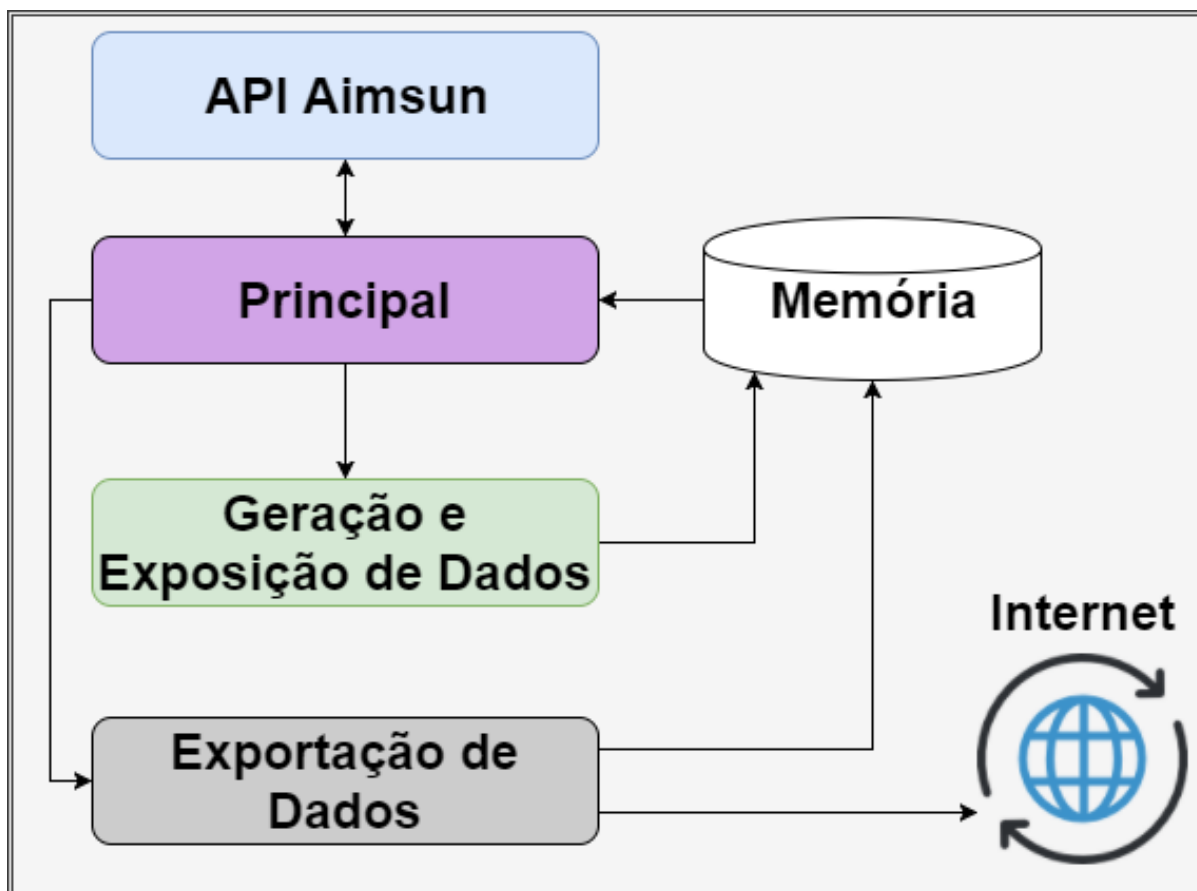
A plataforma de exportação foi construída sobre a estrutura da plataforma de controle previamente desenvolvida [13], e a Figura 3.5 representa a modificação da estrutura base com sua implementação. A plataforma base possuía as funcionalidades de: controle, indicada pelo bloco em cor magenta; e criação de arquivos específicos de demanda e exposição de resultados, indicados pelo bloco de cor verde.

A estrutura de exportação de dados, indicada na cor cinza, é informada pelo *script* ‘Principal’ sobre os dados de importância e pode armazená-los na memória do computador ou, se necessário e possível, enviá-los via internet.

3.2 Módulo de Servidores

O Módulo de Servidores é a seção de trabalho que consiste em aplicações para diferentes plataformas, como Android SO e sistemas Web, que operem em conjunto e em tempo real. O objetivo é criar interfaces claras e objetivas que auxiliem na operação de uma linha de transporte público controlada, e que possam ser usadas pelos usuários de tal linha, seus motoristas e administradores e/ou pelo sistema de trânsito em geral.

Figura 3.5: Estrutura interna da plataforma de controle com a implementação da exportação de dados. O bloco 'Principal' é aquele que se comunica com a API do Aimsun. No projeto atual, o autor desenvolveu o bloco 'Exportação de Dados'.



Fonte: Elaborada pelo autor com base no relatório técnico [13]

3.2.1 Adaptação e uso do Servidor SincroBUS e Atis-Web

O SincroBUS é um sistema dedicado ao desenvolvimento dos protótipos de software, aos serviços Web necessários para a integração dos sistemas existentes e aos clientes web em geral. O aluno fez uso de dois módulos do sistema: o Servidor SincroBUS, ou Cerebus, que serve como mediador e agregador de informações e serviços; e o Atis-Web, um sistema Web voltado aos usuários de transporte público integrado.

O módulo Cerebus é projetado como uma série de recursos Web, cada qual podendo sofrer um conjunto pré-determinado de operações definidas pela RFC 2616, referente ao protocolo HTTP/1.1. Todos os recursos desse módulo estão definidos no manual do sistema [6].

Esse módulo pode receber informações de diversos componentes, como AVL, órgão gestor e outras fontes, e as processar através de seus componentes internos, derivando novas informações. Posteriormente, essas informações permanecem disponíveis como recursos Web, e o seu acesso restrito é permitido perante conjunto de *login* e senha.

Tabela 3.2: URL's usadas pelo Cerebus, seus objetivos de uso e operações HTTP/1.1 suportadas. As operações demarcadas com 'X' são originais do SincroBUS, enquanto 'V' foram criadas pelo autor do PFC e implementadas durante o curso do projeto.

URL	Objetivo de uso	Operações Suportadas			
		POST	PUT	DELETE	GET
/itinerarios	cadastro de itinerários	X			X
/itinerarios/{}	consulta e modificação		V	V	X
.../tabelaDeHorarios	horários do itinerário		X		X
/onibus	cadastro de ônibus	X			X
/onibus/{}	consulta e modificação		V	V	X
.../localizacoes	posição de um ônibus	X			
.../iniciosDeViagem	inicia o uso de um ônibus	X			
.../finsDeViagem	termina o uso de um ônibus	X			
.../recomendacao	recomendação de conduta		V		X
/pontosDeOnibus	cadastro de pontos	X			X
/pontosDeOnibus/{}	consulta e modificação		V	V	X

A Tabela 3.2 apresenta as URL's implementadas no servidor Cerebus e usadas no PFC, seus objetivos de uso e operações HTTP/1.1 suportadas para cada URL. Por exemplo, ao realizar uma operação do tipo 'POST' na URL '/onibus', o servidor utiliza os dados contidos na operação para realizar o cadastro de um novo ônibus no servidor.

As URL's que apresentam o símbolo '{}' em sua expressão esperam, nesse campo, um índice inteiro que represente uma entidade única dentre as cadastradas. Esse índice é escolhido ao realizar o cadastro na entidade. Por exemplo, se um ônibus for cadastrado no sistema com o índice '1', então a URL '/onibus/1/localizacoes' representa a posição desse ônibus.

As URL's que apresentam o símbolo '...' como prefixo são prefixadas pela última coluna acima que não possui tal símbolo em seu início. Seguindo tal regra, a URL da terceira linha é '/itinerarios/{}/tabelaDeHorarios'.

Ao realizar as operações pré-determinadas pelos recursos Web do servidor, o usuário do sistema Web deve enviar um conjunto de dados para o servidor. Esse conjunto de dados

define a entidade, ou o seu uso, relacionada à URL em questão. O tipo do conjunto de dados está definido em [6]. Por exemplo, ao realizar a operação ‘POST’ na URL ‘/onibus’, afim de realizar o cadastro de um novo ônibus, é necessário o envio de um arquivo de texto no formato padrão *Comma-Separated Values* (CSV) contendo os dados necessários para a realização do cadastro. Os tipos de dados usados pelo Cerebus são: textos em formato padrão CSV (geralmente omitindo o cabeçalho); textos simples; e aplicações do tipo *JavaScript Object Notation* (JSON).

As operações demarcadas com o símbolo ‘X’ na Tabela 3.2 são operações originais do SincroBUS. Contudo, essas não foram suficientes para o uso no PFC. Portanto, o aluno requisitou a modificação do servidor para a adição de novas operações em suas URL’s. Tais novas operações estão demarcadas com o símbolo ‘V’. A adição de tais operações tornou o uso do Cerebus muito mais intuitivo e modelável, permitindo testes e adequações quando os resultados dos testes fossem insatisfatórios.

Para adequar o SincroBUS ao cenário construído no Módulo de Simulação, o aluno fez uso das operações suportadas, e apresentadas na Tabela 3.2, afim de adicionar novas informações ao servidor. Essas foram coletadas a partir da simulação através do módulo de exportação criado. As informações coletadas, e posteriormente adicionadas ao servidor, foram: posição dos pontos de ônibus da linha; tamanho de frota usada; pontos pertencentes ao itinerário da linha modelada; e o horário de partida de cada um dos ônibus simulados.

Tabela 3.3: URL’s adicionadas ao Cerebus para implementação de prioridade semafórica e seu efetivo armazenamento de dados.

URL	Objetivo de uso	Operações Suportadas			
		POST	PUT	DELETE	GET
/semaforos	cadastro de semáforos	V			V
/semaforos/{}	consulta e modificação		V	V	V
.../pedidosDePrioridade	prioridades realizadas		V		V

A versão original do SincroBUS foi criada sem a intenção de tratamento de dados e informações sobre os semáforos. Com o objetivo de inserir a capacidade de controlar a malha viária através de prioridades semafóricas, bem como tratar e armazenar dados relativos à essa operação, o autor do PFC idealizou uma modificação do seu servidor

Figura 3.6: Mapa do itinerário construído, como apresentado no sistema Atis-Web.



Fonte: *Print screen* da aplicação, sistema operacional Windows 10

e a requereu sua implementação. As URL's adicionadas, bem como seus métodos implementados, estão apresentados na Tabela 3.3. A estrutura de tal Tabela segue aquela apresentada na Tabela 3.2.

As informações inseridas no Cerebus, bem como as que ele mesmo deriva, são disponibilizadas como recursos para os componentes de observação de dados, como o Atis-Web. Os usuários de transporte público podem usar tal sistema para acompanhar a posição dos

ônibus em tempo real, receber estimativas de tempos de chegada em estações, visualizar o itinerário e a tabela de horário de uma linha em vigência.

Por fim, a Figura 3.6 apresenta como o sistema Atis-Web serve de visualização de rota aos usuários de transporte público. O itinerário, pontos de ônibus e quadro de horários apresentados foram gerados em simulação e importados ao servidor Cerebus através dos recursos Web. No sistema Atis-Web, os usuários de transporte público podem visualizar a movimentação dos ônibus em tempo real, como simulado pelo Aimsun.

3.2.2 Desenvolvimento do Aplicativo Móvel

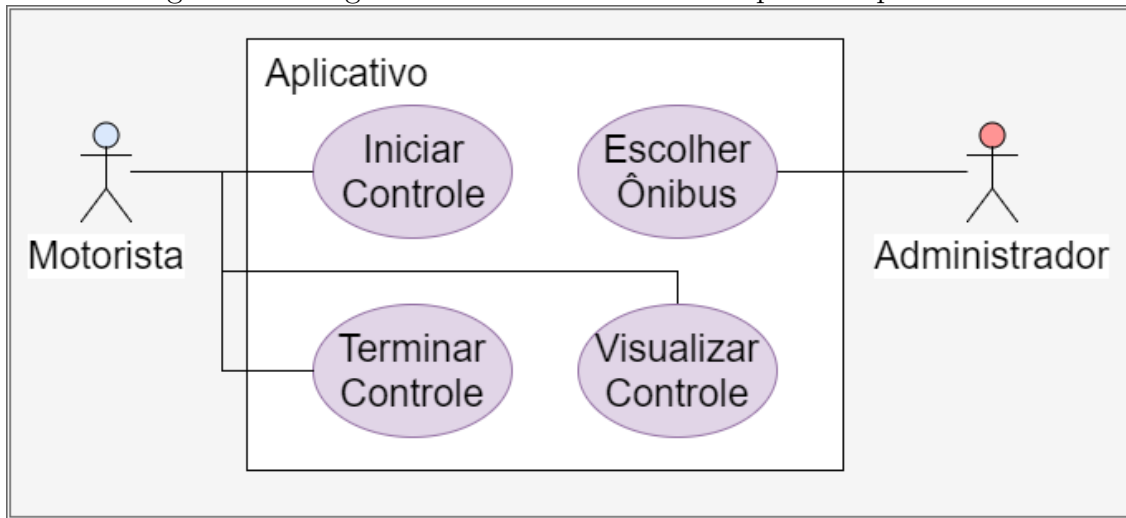
A fim de criar uma interface humano-maquina entre o motorista de um ônibus em uma linha controlada e o sistema de controle, um aplicativo móvel foi considerado pelo autor do PFC e seus orientadores como a melhor plataforma. Isso se deu devido a facilidade de manuseio, atual uso disseminado e ampla confiabilidade dos aparelhos portáteis atuais, como *notebooks* leves, *tablets* e celulares *smartphones*. Todos esses aparelhos podem ser facilmente conectados à Internet e inseridos ao painel de um ônibus.

Para a criação e programação do aplicativo, o autor do PFC fez uso da Kivy [10], uma biblioteca *open-source* para Python. A escolha do uso de tal biblioteca foi realizada devido à facilidade da transição entre a aplicação da plataforma de controle utilizada no projeto e a criação da aplicação móvel. Também de devida importância, Kivy permite a criação de programas multi-plataforma. Portanto, se for realizada uma implementação real, a decisão de qual *hardware* (aparelhos portáteis previamente mencionados) será usado para essa implementação pode ser feita baseando-se em outros fatores, como custos ou adaptabilidade, sem a restrição de uma aplicação de plataforma única.

Para realização de testes no projeto, a aplicação foi criada com objetivo de implementação em um celular *smartphone* com sistema operacional Android. Isso se deu devido ao celular estar à disposição do projeto.

Para auxiliar na compreensão do uso geral do aplicativo, a Figura 3.7 apresenta um Diagrama UML de Casos de Uso para o desenvolvimento do aplicativo. Nele, o Motorista do ônibus e o Administrador da linha de ônibus são atores que agem sobre o aplicativo. Os blocos internos de ‘Aplicativo’ são operações que podem ser realizadas, e as linhas indicam dependência.

Figura 3.7: Diagrama UML de Casos de Uso para o Aplicativo.



Fonte: Elaborada pelo autor

O Motorista, em posse do aplicativo, é capaz de iniciar a operação de controle, visualizar sua indicação e terminar essa operação. Essas operações são realizadas durante a viagem do veículo.

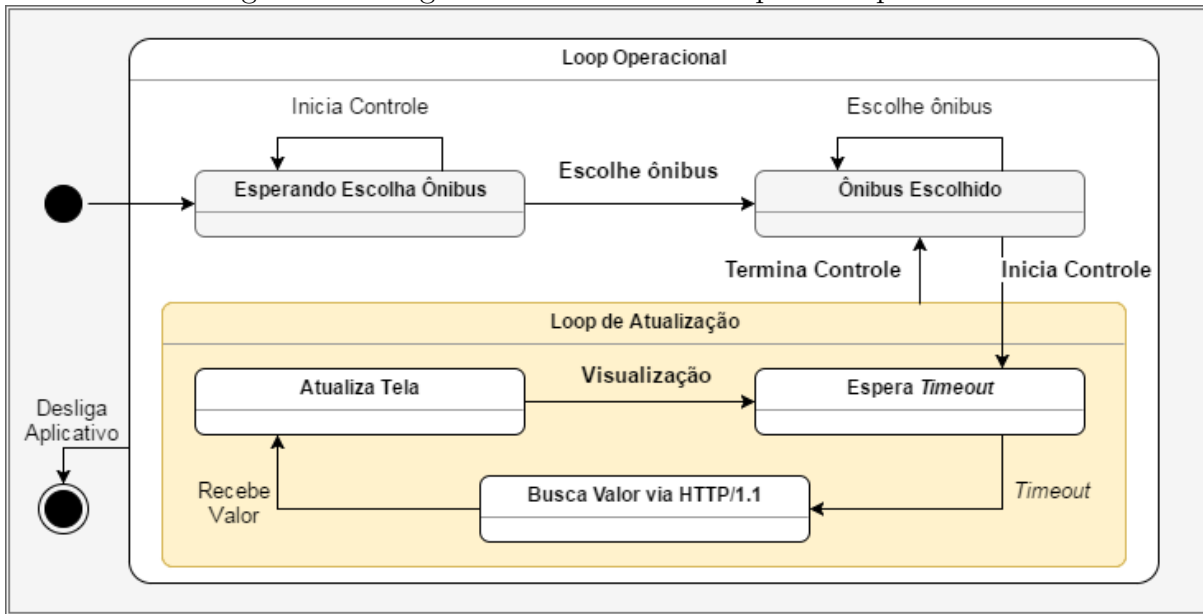
Para garantir que o ônibus seja controlado de acordo com o planejado, o Administrador deve informar ao aplicativo qual é esse ônibus dentre a frota total. Em uma implementação futura e mais abrangente, pode-se informar qual é a linha sendo controlada, características da operação e outras informações que sejam pertinentes.

A Figura 3.8 apresenta um Diagrama UML de Estados para o aplicativo. O ‘Loop Operacional’ representa todo o funcionamento do aplicativo, que só termina quando o usuário o desliga.

A partir do estado inicial, o aplicativo deve esperar a escolha de um ônibus, impedindo o controle de iniciar. Após a escolha efetiva de um ônibus o aplicativo entra um estado de espera para iniciar o controle. Durante tal estado, é possível escolher outro ônibus, se desejado. Caso inicie o controle, o aplicativo passa para o ‘Loop de Atualização’, onde os três próximos estados são realizados em sequência. Esse loop pode ser interrompido através da ação de terminar o controle, onde o aplicativo volta ao estado de ônibus definido.

O estado ‘Espera *Timeout*’ é onde se define o período de atualização da imagem que indicará ao motorista qual é a recomendação atual. Ao ocorrer o evento ‘Timeout’, avançar-

Figura 3.8: Diagrama UML de Estados para o Aplicativo.



Fonte: Elaborada pelo autor

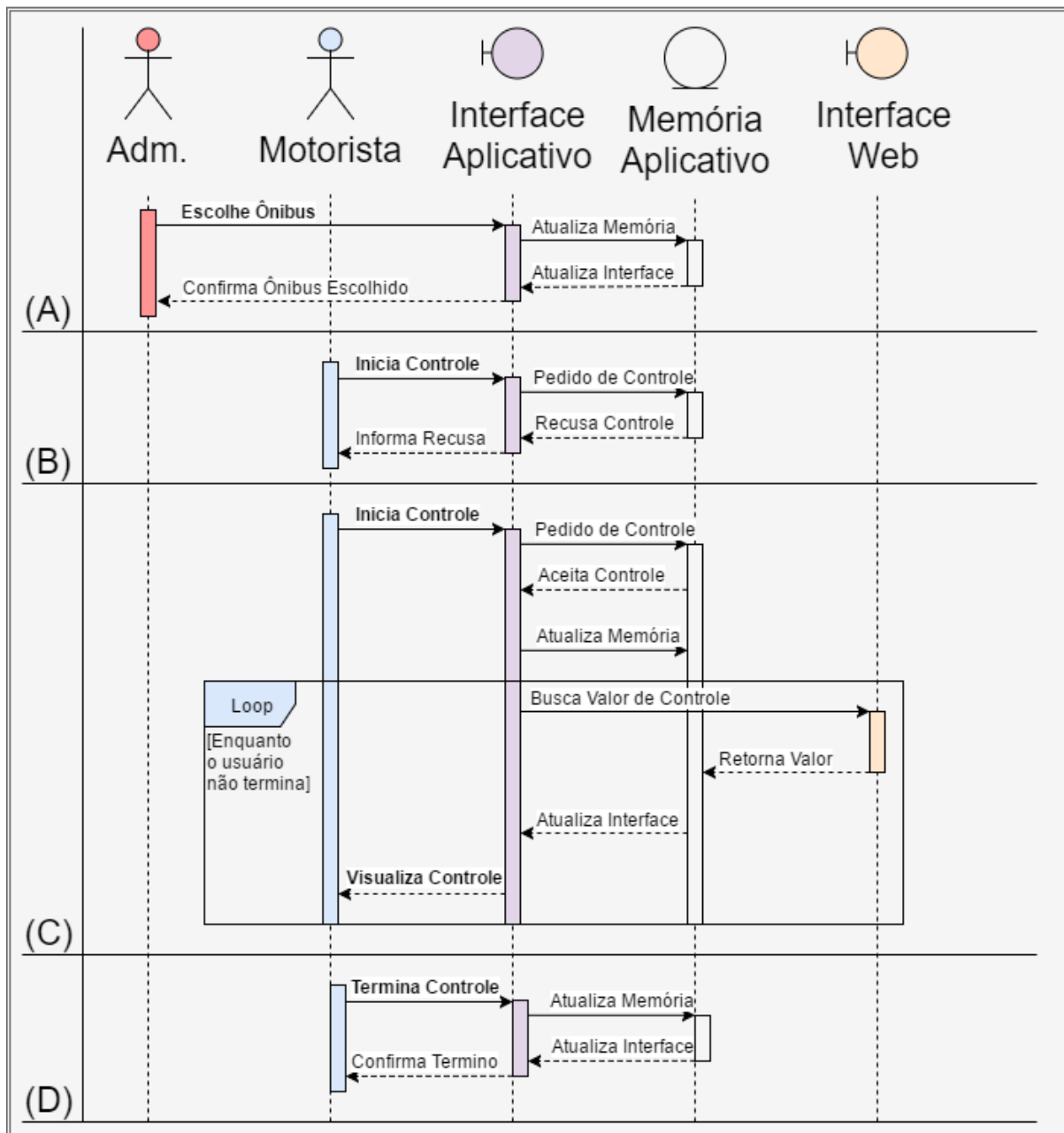
se de estado. ‘Busca Valor via HTTP/1.1’ é o estado onde a recomendação de conduta é observada através de uma interface web. A recomendação está no servidor Cerebus como um recurso web. Ao ocorrer o evento ‘Recebe valor’, passa-se para o próximo estado. ‘Atualiza Tela’ é o estado onde a indicação de recomendação é dada ao motorista. O evento ‘Visualização’ é uma abstração a ação do motorista; o aplicativo assume que o motorista está atento à tela e segue instantaneamente para o próximo estado.

A Figura 3.9 apresenta um Diagrama UML de Sequência para as operações do aplicativo. As sequências temporais de cada operação estão indicadas na imagem, margem esquerda, em ordem alfabética. Respectivamente, são elas: (A) Escolher um ônibus; (B) Iniciar Controle, operação recusada (como ocorre no estado ‘Esperando Escolha Ônibus’); (C) Iniciar Controle, operação aceita e visualização do Motorista; e (D) Terminar Controle.

A operação Escolher Ônibus, indicada pela sequência (A), é a única realizada pelo Administrador. Esse ator age sobre a interface do aplicativo, indicando qual ônibus será controlado. A memória do aplicativo é atualizada, a interface recebe a confirmação de definição do ônibus, e o Administrador confirma a escolha.

A operação Iniciar Controle, quando realizada sem o ônibus escolhido pelo Administrador, é indicada pela sequência (B). Nela, o Motorista age sobre a interface, indicando

Figura 3.9: Diagrama UML de Sequência para as operações do aplicativo o Aplicativo. As operações são: (A) Escolher um ônibus; (B) Iniciar Controle operacional, mas início recusado; (C) Iniciar Controle operacional, com visualização do Motorista; e (D) Terminar Controle.



Fonte: Elaborada pelo autor

o início do controle. A interface realiza o pedido de início da operação de controle, mas a memória recusa esse pedido, devido a falta de qual ônibus deve ser controlado. A interface é atualizada e o Motorista é informado da recusa.

A operação Iniciar Controle, quando realizada de forma correta, é indicada pela sequência (C). O Motorista age sobre a interface, indicando o início do controle. A interface realiza o pedido de início da operação de controle, e a memória aceita o pedido de controle. A memória é atualizada, indicando o início do controle, e o sistema inicia um ‘loop’. Em sequência, as ações de: buscar o valor de controle via Web; receber tal valor; e atualizar a interface do aplicativo são realizadas, com período cíclico definido. Assim, o Motorista pode Visualizar Controle, indicado na interface do aplicativo.

Caso desejado, o Motorista pode realizar a operação de Terminar Controle, indicado pela sequência (D). O Motorista age sobre a interface, indicando o término do controle. A memória é atualizada, indicando o término do controle, e a interface é atualizada, impedindo o ‘loop’ operacional de controle. Por fim, o Motorista confirma o término de controle.

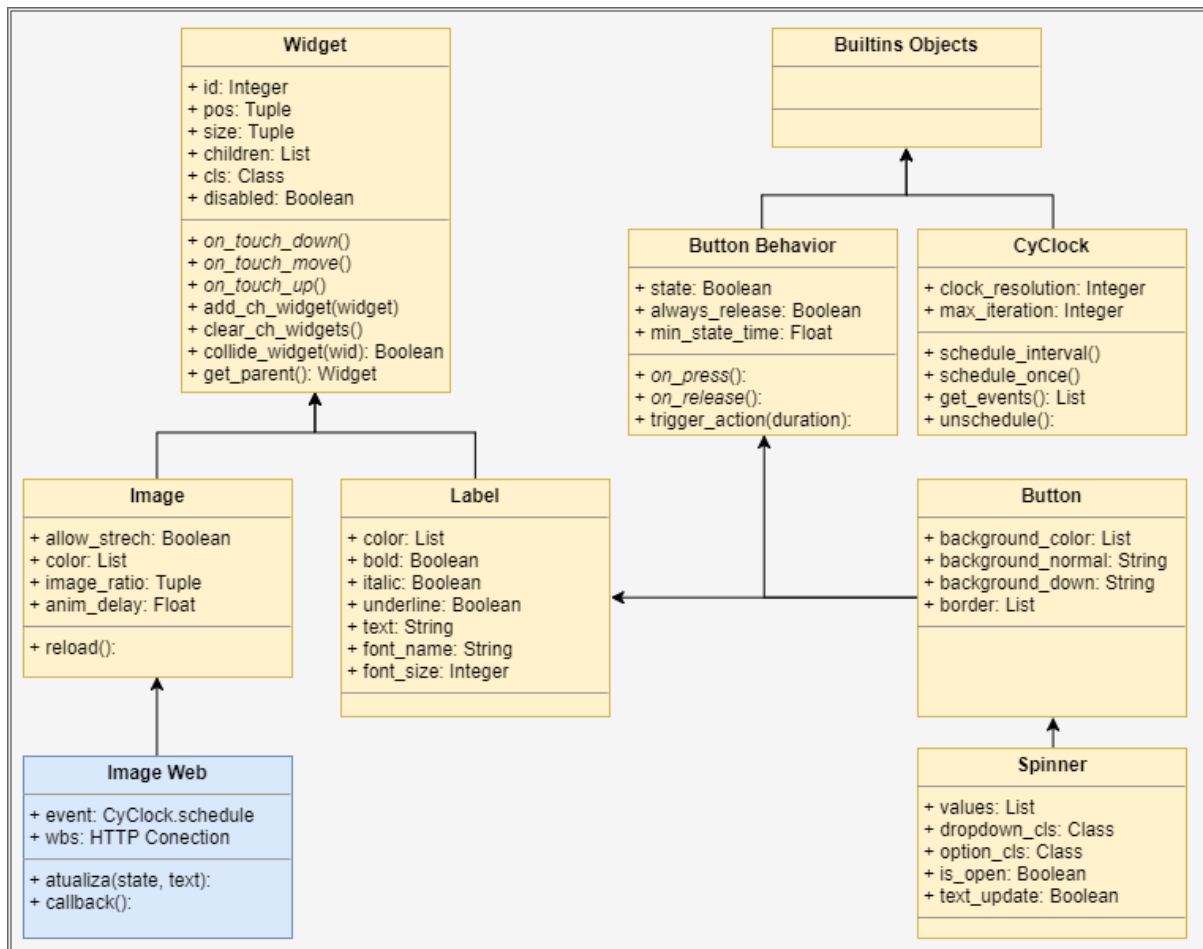
A Figura 3.10 apresenta um Diagrama UML de Classes representando as estruturas Kivy utilizadas na criação do aplicativo móvel. Existem outras classes na biblioteca, mas não foram usadas no projeto ou não são o foco de estudo no projeto, como as classes ‘Screen’ ou ‘Screen Manager’.

As setas representam hierarquia, característica básica de linguagens de programação de alto nível, onde as classes apontadas são super classes. Por exemplo, ‘Button’ estende ‘Label’, que por sua vez estende ‘Widget’. Essa última classe é uma das básicas da biblioteca Kivy, geralmente considerada a raiz de todas as outras estruturas.

As classes indicadas na cor amarela são classes base da biblioteca Kivy. Por outro lado, a classe ‘Image Web’, indicada em azul, foi construída pelo autor do PFC. Isso se deu devido a necessidade de uma imagem que possua intrínseca conexão com a internet. Essa conexão é instanciada durante a operação do aplicativo através da biblioteca ‘HTTPLib’, do Python.

Os métodos escritos em itálico são baseados em eventos, e estão todos diretamente relacionados com a capacidade da biblioteca Kivy interpretar ações do usuário, sejam elas via toque (*smartphones* ou *tablets*), ou mouse e teclado (computadores comuns). A classe ‘Builtins Objects’ apresenta atributos e métodos em demasia, impossibilitando a escrita destes em uma imagem sucinta.

Figura 3.10: Diagrama UML de Classes representando as estruturas Kivy utilizadas na criação do aplicativo móvel.



Fonte: Elaborada pelo autor

Após estudo realizado sobre a biblioteca e suas classes, o autor do PFC optou por implementar as operações desejadas de maneira que:

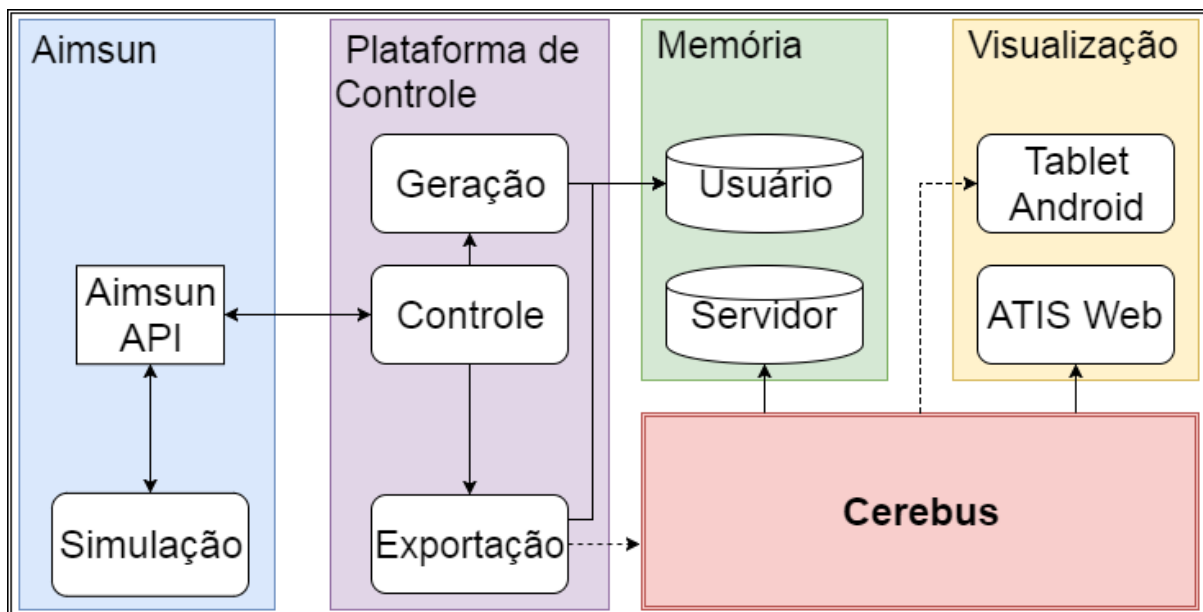
- Iniciar e Terminar Controle: instância de ‘Button’ simples, onde o seu atributo herdado ‘always_release’ é sempre ‘False’, construindo um botão que troca de estado ao ser pressionado, mas permanece nesse estado caso o usuário o solte. Caso pressionado, o Controle está ativado; caso contrário, o Controle está desativado.
- Escolher ônibus: instância de ‘Spinner’ simples, onde seu atributo ‘values’ é uma lista contendo todos os possíveis ônibus a ser controlados. Possibilita a redefinição do ônibus escolhido.

- Visualizar Controle: instância de ‘Image Web’, classe derivada de ‘Image’. Caso o Controle esteja ativo, essa classe cria uma conexão com a web e busca o valor de controle armazenado na memória do SincroBUS. Depois, atualiza sua imagem adequando-a a situação atual.

3.2.3 Arquitetura Global do Sistema Computacional

Após o desenvolvimento de todos os Módulos e suas Etapas, é necessário que todos os componentes sejam capaz de enviar e receber dados de maneira simples e em tempo real, para que a função de todos os softwares seja cumprida. A Figura 3.11 apresenta um diagrama que representa a arquitetura global de relações entre todos os componentes do sistema computacional. Essa figura é uma representação mais específica e detalhada daquela apresentada na Figura 3.1.

Figura 3.11: Arquitetura final do sistema computacional projetado. As setas representam o fluxo de informação. As setas tracejadas simbolizam conexão via web.



Fonte: Elaborada pelo autor

O bloco em azul, ‘Aimsun’, representa o Módulo de Simulação e seus componentes. A Simulação da malha viária e da linha de transporte público modelada se comunica com sua API (*Application Program Interface*), que por sua vez realiza uma troca de dados com a ‘Plataforma de Controle’, indicada pelo bloco roxo.

Essa Plataforma representa a capacidade do sistema de interpretar os dados do ambiente simulado e calcular ações sobre cada veículo da linha modelada. Essas ações retornam à API e são realizadas na simulação, controlando o sistema. Por outro lado, os dados observados da simulação e seus derivados podem ser exportados em formato padrão, como um sistema AVL, capacidade indicada pelo sub-bloco ‘Exportação’. O bloco ‘Geração’ indica a capacidade da plataforma de controle de gerar arquivos variados, necessários para a operação e análise do sistema.

A partir do sub-bloco ‘Exportação’ partem duas setas. A seta contínua entre esse e o bloco verde, ‘Memória’, indica a capacidade da Plataforma de criar arquivos contendo dados da simulação e arquivá-los na Memória do computador onde a simulação é realizada. Tais arquivos podem ser *feeds* GTFS ou lotes CSV para o SincroBUS.

A seta pontilhada entre o sub-bloco ‘Exportação’ e o bloco vermelho, ‘Cerebus’, indica uma conexão via Internet. Essa conexão é realizada em tempo real em relação ao que é observado na simulação. Os dados enviados simulam informações AVL, como as posições de cada veículo ao longo do tempo, e indicações de recomendação de conduta calculadas se o controle for por meio da velocidade.

O bloco ‘Cerebus’ serve como um agregador e organizador de dados para os módulos que os requerem. O servidor pode armazenar dados em sua memória, capacidade indicada pela seta entre o servidor e o sub-bloco ‘Servidor’ de ‘Memória’ para que quando desejado esses dados sejam visualizados por aqueles que possuem acesso ao servidor.

O bloco amarelo, ‘Visualização’, representa todas as interfaces construídas ou adaptadas para a observação de dados da simulação, de acordo com o Módulo de Servidores.

O ATIS Web, do SincroBUS, é indicado aos usuários do sistema de transporte público, e apresenta de maneira intuitiva a posição dos veículos no mapa da linha, as estações de embarque e desembarque, horários de saída dos ônibus do terminal central, etc.

O sub-bloco ‘Tablet Android’ indica a interface entre o motorista do veículo da linha controlada e o sistema de controle, com indicações simples de aceleração/desaceleração para o seu ônibus ao longo do tempo, para que seja atingido o objetivo de controle. A comunicação do Tablet e o Cerebus é feita via Internet, protocolo HTTP/1.1, como indicado pela seta pontilhada. Essa comunicação é discreta, realizada com período determinado, e depende da pré-definição, no aplicativo, de qual ônibus será representado pelo Tablet.

4 Desenvolvimento Matemático do Controle Híbrido

O Módulo de Controle é a seção do trabalho que comporta os algoritmos de maior complexidade matemática, responsáveis por tomar as decisões de controle. O objetivo desse controle é relacionar os dados de campo e encontrar os acionamentos sobre a operação da linha de transporte público. Deve-se normalizar os intervalos observados de N ônibus de uma mesma linha. Os dados de campo são provenientes da simulação construída pelo autor, apresentada na Seção 3.1, e considera informações da malha viária e dos outros ônibus da linha. O acionamento é realizado através da regulação da velocidade média do ônibus em cada trecho entre estações. Essa regulação é realizada através de recomendações de conduta ao motorista.

O controle utilizado nesse trabalho é considerado um Controle Híbrido, pois possui uma parcela orientada por eventos e outra discreta. Um evento ocorre toda vez que um ônibus chega em uma estação de embarque e desembarque, onde a velocidade média para o próximo trecho é calculada. Paralelamente, um controle discreto é acionado a todo instante em que o ônibus não está em um ponto de embarque e desembarque, com intenção de manter a velocidade média calculada, indicando recomendações simples e diretas ao motorista.

A nomenclatura de variáveis utilizadas nesse texto segue aquela usada em [2], com exceção de $w_{i,k}$. Essas variáveis, dada uma estação k e um ônibus i , estão apresentadas na Tabela 4.1. Os dados estão em suas unidades padrão, metros para distância e segundos para tempo.

Tabela 4.1: Variáveis usadas no texto e seus significados.

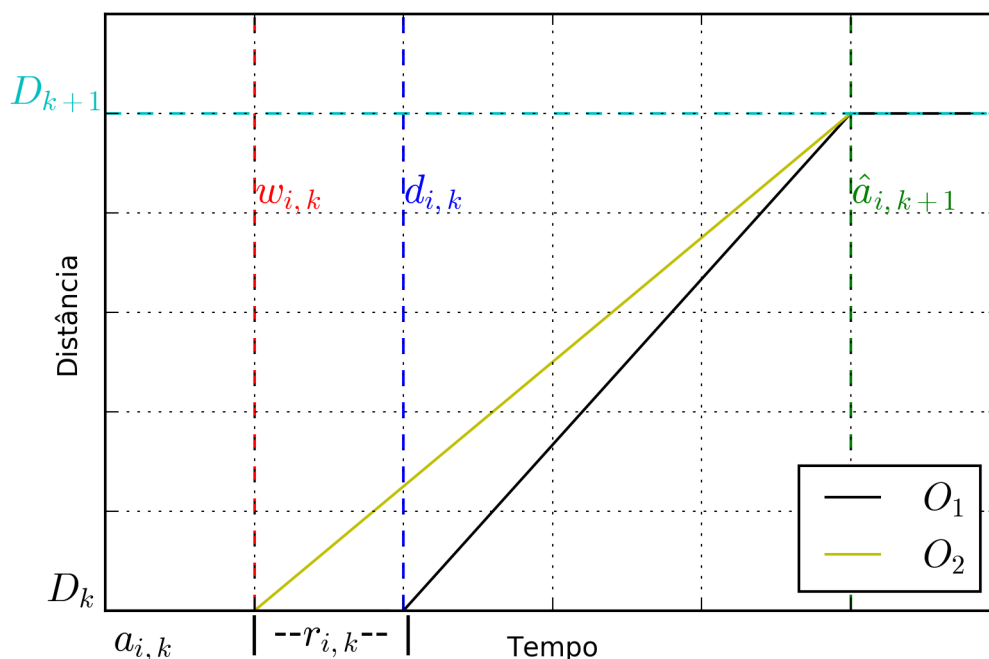
Variável	Significado
D_k	Distância da origem até a estação k
$a_{i,k}$	Instante de chegada do ônibus i na estação k
$\hat{a}_{i,k}$	Instante previsto de chegada no ônibus i na estação k
$w_{i,k}$	Instante de embarque/desembarque completo do ônibus i na estação k
$d_{i,k}$	Instante de partida do ônibus i na estação k
$r_{i,k}$	Tempo de retenção do ônibus i na estação k
$t_{i,k}$	Tempo de viagem do ônibus i no trecho entre a estação k e a próxima

4.1 Controle a Eventos

Na ocorrência do evento de um ônibus chegar à uma estação, a plataforma de controle por retenção é responsável por concluir qual é o tempo de retenção que deve ser aplicado ao ônibus para aumentar a regularidade do sistema. A intenção do uso dessa retenção é garantir que um ônibus i chegue na próxima estação no instante previsto e normalize o intervalo em relação ao ônibus $i - 1$. Como todos os N ônibus da linha são controlados, então todos os intervalos são regularizados.

Durante o curso do projeto foi realizada uma adaptação da plataforma de controle por retenção, a fim de se adequar aos objetivos iniciais. O objetivo da adaptação realizada em relação ao Controle a Eventos é transformar a informação de retenção em uma redução na velocidade média do veículo a ser realizada.

Figura 4.1: Veículos O_1 e O_2 e suas velocidades ao longo do tempo. O veículo O_2 (amarelo) não possui retenção aplicada, e parte da estação k quando processo de embarque/desembarque estiver completo. As linhas tracejadas verticais indicam: instante de embarque/desembarque completo (vermelho); instante de partida para sistema com retenção (azul); instante previsto de chegada na estação $k + 1$ (verde). A linha tracejada horizontal indica a distância da origem até a estação $k + 1$ (ciano).



Fonte: Elaborada pelo autor

Dados dois ônibus, O_1 e O_2 , sendo O_1 com retenção aplicada, a Figura 4.1 apresenta uma previsão justa das suas posições ao longo do tempo. Aqui, é considerado que O_2 não é retido, mas chega na próxima estação, junto com o primeiro veículo, no instante previsto de chegada. Claramente o veículo não retido possui uma redução em sua velocidade média. Ao fazer uso de tal velocidade média, O_2 atinge o objetivo de controle, ou seja, chegar à estação $k + 1$ no instante previsto $\hat{a}_{i,k+1}$. Portanto, essa velocidade será daqui por diante chamada *Velocidade Média de Referência*, ou \bar{V}_r , definida por:

$$\bar{V}_r = \frac{D_{k+1} - D_k}{t_{i,k} + r_{i,k}} \quad (1)$$

Ao realizar a implementação do Controle a Eventos, a equação acima deve ser calculada toda vez que um ônibus i chegar em uma estação k , e o valor de \bar{V}_r será posteriormente usado pelo Controle Discreto.

Um sistema de controle de intervalos que usa o acionamento por retenção não pode reter um veículo por tempos negativos. Portanto, $r_{i,k} \geq 0$ para todas as estações k e ônibus i quando controle por retenção é usado. Isso se deve ao fato que, ao partir um ônibus antes do tempo previsto de embarque e desembarque de passageiros, a operação do transporte público se torna contra-produtiva e, possivelmente, abominável pelos seus usuários.

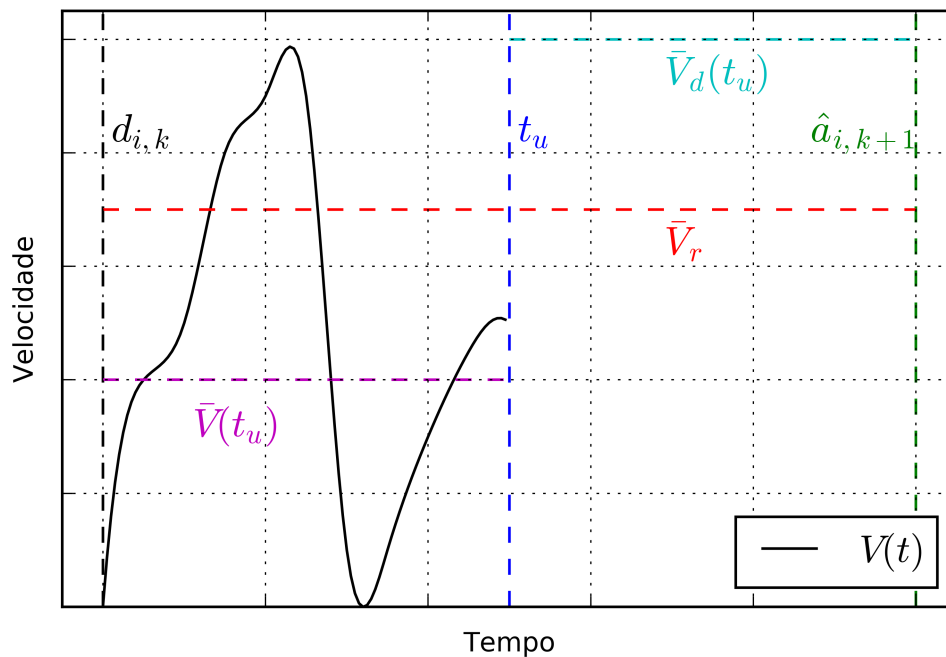
Contudo, a implementação do cálculo da \bar{V}_r não apresenta tal restrição. Ao fazer uso de $r_{i,k} < 0$, causa-se apenas um aumento no valor da velocidade média do veículo. A questão de que se o veículo é capaz ou não de atingir tal velocidade é tratada no Controle Discreto.

Caso $r_{i,k} \leq 0$ e $|r_{i,k}| \geq |t_{i,k}|$ o uso da equação 1 causa instabilidade e erros nos resultados obtidos. Porém, sob operações normais, tráfego não saturado e alta previsibilidade do tempo de viagem, $r_{i,k}$ possui uma ordem menor do que $t_{i,k}$, evitando tais instabilidades. Em uma aplicação robusta, com intenção de aplicação em ambientes saturados e com baixa previsibilidade, essa relação deve ser verificada antes do cálculo.

4.2 Controle Discreto

O Controle Discreto é acionado de modo periódico e apenas para os ônibus que não estão parados em pontos de embarque e desembarque. O objetivo de tal controle é informar ao motorista qual é a velocidade que ele deveria manter naquele instante, a fim de normalizar os intervalos entre estações.

Figura 4.2: Um veículo e sua velocidade ao longo do tempo. As linhas tracejadas indicam: velocidade média do veículo no instante t_u (magenta); velocidade desejada do veículo no instante t_u (ciano); velocidade média de referencia (vermelho); instante de partida do veículo na estação k (preto); instante de controle (azul); instante previsto de chegada do veículo na estação $k + 1$ (verde).



Fonte: Elaborada pelo autor

Dado um veículo e sua velocidade realizada $V(t)$, em um instante de tempo de controle t_u e sua velocidade média realizada $\bar{V}(t_u)$, deve-se encontrar um valor constante de *Velocidade Média Desejada*, ou \bar{V}_d , que deve ser mantido pelo veículo a partir do instante t_u para atingir uma determinada \bar{V}_r no trecho.

Considerando o exemplo demonstrado na Figura 4.2, pode-se observar os instantes de interesse: $d_{i,k}$, em tracejado preto; t_u , em tracejado azul; e $\hat{a}_{i,k+1}$, em tracejado verde. As variáveis de interesse, dado o instante t_u , são: $\bar{V}(t_u)$, em tracejado magenta; e $\bar{V}_d(t_u)$, o

dado que tem-se interesse em encontrar, em tracejado ciano. \bar{V}_r , em tracejado vermelho, é definida para todo o trecho de viagem.

Por definição de velocidade média, a distância $D_{i,k}$ percorrida por um veículo i após a partida da estação k , caso viaje em sua velocidade V ou constantemente em sua velocidade média \bar{V} , é a mesma. Portanto:

$$\bar{V}(t_u) \cdot [t_u - d_{i,k}] = \int_{d_{i,k}}^{t_u} V(t) dt = D_{i,k}(t_u) \quad (2)$$

Vale a pena indicar que, como esse veículo não é retido, seu instante de partida é igual ao seu instante de embarque e desembarque, ou seja, $d_{i,k} = w_{i,k}$. Portanto, tempo previsto de viagem para esse veículo é $t_{i,k} = \hat{a}_{i,k+1} - d_{i,k} = \hat{a}_{i,k+1} - w_{i,k}$.

No projeto, o primeiro ônibus da linha serve para sensoriamento da malha viária durante a simulação. O instante previsto de chegada em cada estação é definido pelo tempo de viagem desse primeiro ônibus. Em uma implantação real, contudo, esse valor deve considerar a malha viária como um todo, incluindo hora do dia e outros possíveis fatores que influenciam no tempo de viagem efetivo.

Também é importante indicar que \bar{V}_d nunca deve extrapolar as velocidades máxima e mínima da via. Tais velocidades serão denotadas $V_{max}(t)$ e $V_{min}(t)$, respectivamente. São expressadas em função do tempo uma vez que o veículo pode estar em ruas diferentes, com diferentes velocidades mínimas e máximas, em instantes diferentes. Usualmente, no Brasil, a velocidade mínima de uma via é definida pela metade da velocidade máxima, e esse é o padrão utilizado no trabalho.

O objetivo garantir que a velocidade média realizada \bar{V} e a velocidade média de referência \bar{V}_r no instante $\hat{a}_{i,k+1}$ sejam iguais. Esse objetivo é atingido modificando a velocidade média desejada \bar{V}_d do veículo ao longo do tempo. Portanto, a média entre $\bar{V}(t_u)$ e $\bar{V}_d(t_u)$ ponderada pelo tempo t_u deve ser:

$$\frac{\bar{V}(t_u) \cdot [t_u - d_{i,k}] + \bar{V}_d(t_u) \cdot [\hat{a}_{i,k+1} - t_u]}{\hat{a}_{i,k+1} - d_{i,k}} = \bar{V}_r \quad (3)$$

Deseja-se definir uma equação que determine um valor de \bar{V}_d para cada instante de controle t_u durante a viagem de duração prevista $t_{i,k}$. Portanto, a partir das equações (3) e (2), tem-se:

$$\int_{d_{i,k}}^{t_u} V(t) dt + \bar{V}_d(t_u) \cdot [\hat{a}_{i,k+1} - t_u] = \bar{V}_r \cdot t_{i,k} \quad (4)$$

$$\bar{V}_d(t_u) \cdot [\hat{a}_{i,k+1} - t_u] = \bar{V}_r \cdot t_{i,k} - \int_{d_{i,k}}^{t_u} V(t) dt \quad (5)$$

$$\bar{V}_d(t_u) = \left[\bar{V}_r \cdot t_{i,k} - \int_{d_{i,k}}^{t_u} V(t) dt \right] \cdot [\hat{a}_{i,k+1} - t_u]^{-1} \quad (6)$$

Generalizando para todo $t \in [d_{i,k}, \infty)$ e considerando as limitações de velocidades máxima e mínima da via, temos:

$$\bar{V}_d(t) = \begin{cases} \{[\bar{V}_r \cdot t_{i,k} - D_{i,k}(t)] \cdot [\hat{a}_{i,k+1} - t]^{-1}\}^{sat}, & \text{se } d_{i,k} \leq t \leq \hat{a}_{1,k+1} \\ V_{max}(t), & \text{se } t \geq \hat{a}_{1,k+1} \end{cases} \quad (7)$$

Sendo

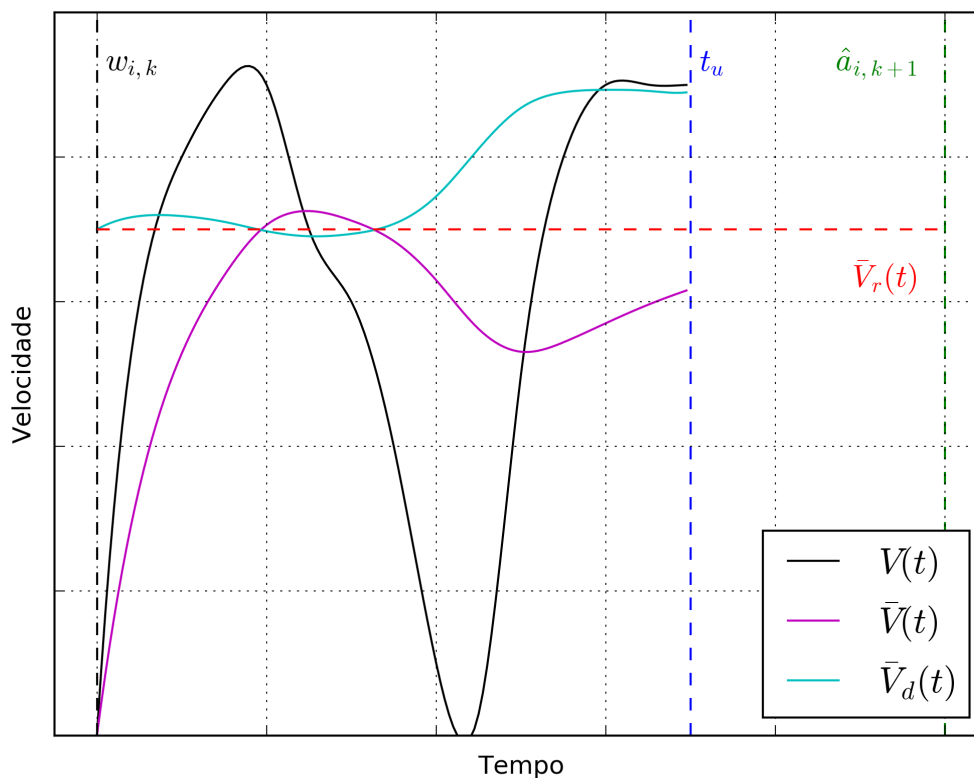
$$x(t)^{sat} = \begin{cases} V_{min}(t), & \text{se } V_{min}(t) \geq x(t) \\ x(t), & \text{se } V_{min}(t) < x(t) < V_{max}(t) \\ V_{max}(t), & \text{se } x(t) \geq V_{max}(t) \end{cases} \quad (8)$$

A equação (7) representa a expressão usada para concluir qual é a velocidade média desejada em instante de tempo t para o veículo. A equação (8) representa a saturação do valor da velocidade média desejada em função dos valores mínimos e máximos da via.

A generalização do tempo pode ser realizada, uma vez que todo instante de tempo $t_u < d_{i,k}$ é anterior a partida do ônibus da estação k , e portanto ele está sendo controlado em função da velocidade média de referência da estação $k - 1$. Se $t_u \geq a_{i,k+1}$, ou seja, o ônibus não chegou à estação $k + 1$ no tempo previsto, então o ônibus está demasiado atrasado e deve viajar na maior velocidade possível.

Importante notar que a equação (7) está em tempo contínuo, mas a implementação é discreta. Portanto, deve-se encontrar um período de amostragem eficiente para cada situação, levando em consideração os menores tempos previstos de viagem, a quantidade de interferência do cenário e a dinâmica de aceleração prevista dos ônibus.

Figura 4.3: Um veículo e suas velocidades real (preto), média (rosa) e média desejada (ciano) ao longo do tempo. As linhas tracejadas indicam: velocidade média de referencia (vermelho); instante de partida do veículo na estação k (preto); instante de controle (azul); instante previsto de chegada do veículo na estação $k + 1$ (verde).



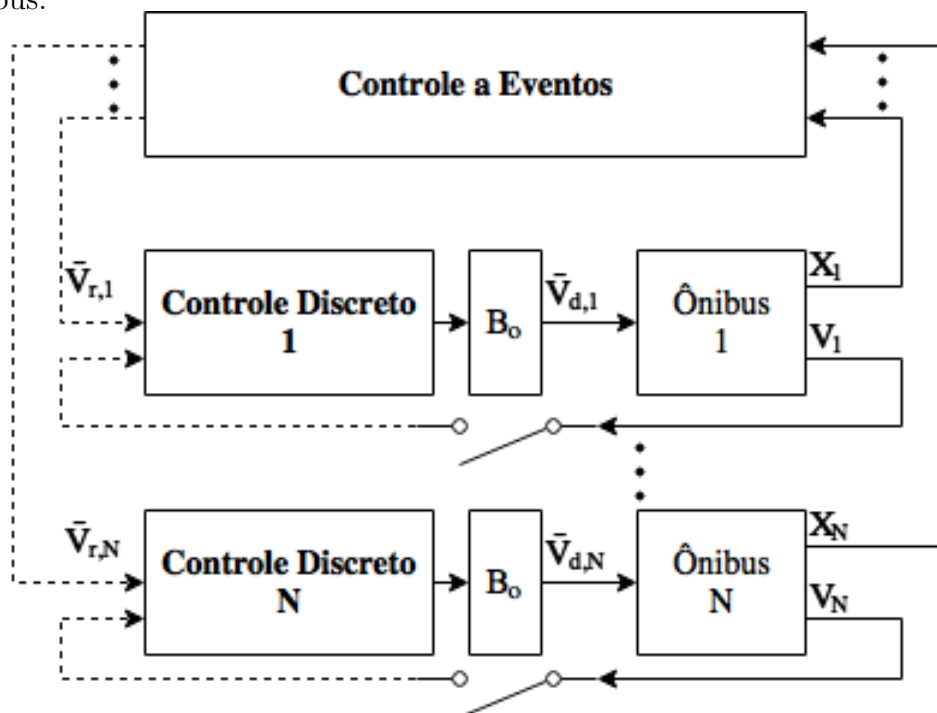
Fonte: Elaborada pelo autor

Na Figura 4.3, podemos observar uma simulação da aplicação da equação (7) em um ambiente sem atingir a saturação de velocidades máxima. A velocidade real do veículo, indicada em preto, gera a velocidade média ao longo do tempo, em magenta. O valor da velocidade média desejada, indicada em ciano, é aquela que o motorista do veículo deve se basear para atingir o objetivo de controle. Fica claro que, ao diminuir a velocidade média sendo realizada na porção mediana do trecho, o valor da velocidade média desejada aumenta, a fim de se atingir o objetivo de controle.

4.3 Relação entre Controles e Criação do Controle Híbrido

A Figura 4.4 apresenta um diagrama de blocos da relação entre os Controles construídos e os ônibus em operação. Considera-se que cada ônibus i e seu respectivo motorista é um sistema que observa $\bar{V}_{d,i}$ como uma velocidade ideal de referência e age no ambiente sob a influência de perturbações e dinâmicas naturais. Isso manipula a velocidade do ônibus (V_i) e, conseqüentemente, sua posição (X_i).

Figura 4.4: Diagrama de Blocos da relação entre Controle a Eventos, Controle Discreto e N Ônibus.



Fonte: Elaborada pelo autor

Tais informações são usadas, respectivamente, pelo Controle Discreto de cada ônibus e pelo Controle a Eventos. Cada ônibus possui seu Controle Discreto próprio, enquanto o Controle a Eventos é único e usa informações de mais de um ônibus para concluir qual deve ser a $\bar{V}_{r,i}$ de cada ônibus i da malha.

Os blocos ‘Segurador de Ordem Zero’ (Bo) e ‘Amostrador’ (chave) servem para dar consistência matemática ao diagrama, devida natureza contínua da operação, mas discreta dos Controles Discretos. O Controle a Eventos observa continuamente os ônibus, manipulando sua interpretação interna de estados, mas os valores de $\bar{V}_{r,i}$ de cada ônibus pode ser observado de modo discreto pelos controladores individuais.

4.4 Quantização da Indicação de Velocidade Desejada

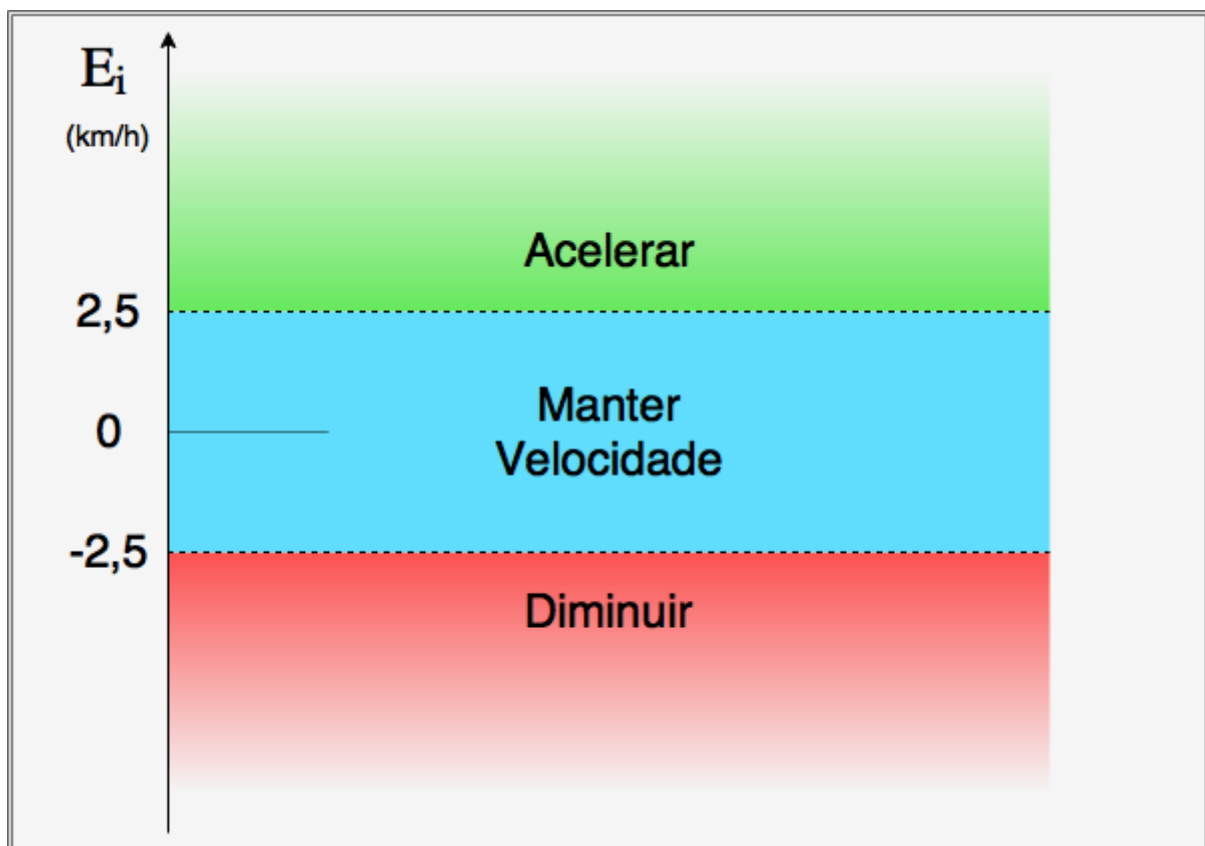
O objetivo do sistema computacional, em relação à Velocidade Desejada, é indicar ao motorista orientações simples sob a forma de variações em aceleração. O SincroBUS foi construído para armazenar e distribuir dados de recomendação de conduta, sendo que sua representação interna de tal recomendação é uma *string* podendo assumir um de quatro valores, significando:

1. ‘Manter Velocidade’: quando o ônibus não deve alterar sua velocidade;
2. ‘Acelerar’: quando o ônibus está com a velocidade atual abaixo da desejada;
3. ‘Diminuir’: quando o ônibus está com a velocidade atual acima da desejada;
4. ‘Sem Dados’: usado quando o sistema de controle não possui indicações de velocidade para o instante atual.

Esse padrão é mantido durante todo o projeto. É necessário encontrar, no ônibus i , o erro de velocidade atual $E_i(t) = \bar{V}_{d,i}(t) - V_i(t)$ e, subsequentemente, quantiza-lo entre as 4 formas de recomendação implantadas pelo SincroBUS. Durante o projeto, as faixas de operação de $E_i(t)$ foram implementadas de acordo com o apresentado na Figura 4.5. A indicação ‘Sem Dados’ não está apresentada pois sua implementação não depende de $E_i(t)$.

Importante notar que a quantização não é realizada ao influenciar as ações dos motoristas simulados no Aimsun. Estes recebem o valor ideal de velocidade média e tentam alcançá-la ao decorrer do tempo. O valor quantizado é usado apenas na aplicação móvel desenvolvida no projeto. Estudos futuros são necessários para observar a reação de um motorista real à informação de recomendação quantizada.

Figura 4.5: Faixas de operação do Erro de Velocidade E_i quantizadas em três recomendações diferentes: ‘Acelerar’, em verde; ‘Manter Velocidade’, em azul; e ‘Diminuir’, em vermelho.



Fonte: Elaborada pelo autor

5 Resultados

Nessa seção são apresentados os resultados atingidos no PFC. Foi considerado que não existe necessidade da apresentação de resultados para o Módulo de Simulação, uma vez que todos os resultados do Módulo de Controle são gerados a partir da simulação construída.

5.1 Módulo de Controle

Os resultados obtidos do Módulo de Controle e sua adaptação do controle por retenção para controle por velocidade são divididos em duas seções: aqueles em que o cenário simulado é completo; e aqueles onde existe prioridade semafórica total para ônibus. Isso se dá para teste de hipótese de que a presença de semáforos é prejudicial ao controle por velocidade.

5.1.1 Cenário Tradicional

A plataforma de controle original usada no projeto possuía acionamento por retenção. Durante o curso do PFC, o autor realizou a adaptação de tal plataforma para acionamento por velocidade. Para comprovar que a adaptação da plataforma de controle é eficiente, possuindo resultados previsíveis e comparáveis ao original, um total de trezentos e cinquenta e cinco (355) observações de intervalos simulados foi realizada. Busca-se observar o intervalo $h_{i,k}$, definido como a diferença dos valores $d_{i,k}$ e $d_{i+1,k}$, entre todos os ônibus de uma linha em todas as estações, uma vez que o objetivo de controle é normalizar tal valor.

Essas observações foram separadas igualmente em dois grupos de acordo as suas diferenças em relação ao método de acionamento: Grupo de Controle (*GC*), simulações com acionamento por retenção; e Grupo de Teste (*GT*), simulações com acionamento por velocidade. A Tabela 5.1 apresenta o resumo de tais grupos, contendo sua nomenclatura usada, o tipo de acionamento, e o número de observações em cada grupo.

Importante notar que, mesmo nas simulações realizadas com acionamento por velocidade, o valor do instante de partida $d_{i,k}$, quando usado para concluir $h_{i,k}$, leva em consideração a retenção $r_{i,k}$ calculada na estação. Isso se dá por que, mesmo que tal re-

tenção não seja aplicada ao ônibus, o objetivo de Controle Discreto é que a velocidade média do veículo seja reduzida e ele chegue na estação $k + 1$ com o atraso relativo à $r_{i,k}$, atingindo o intervalo desejado.

Tabela 5.1: Características dos grupos de observações realizadas, contendo suas nomenclaturas, tipos de acionamento e número total de observações.

Grupo	Acionamento	Nº de Observações
<i>GC</i>	Retenção	186
<i>GT</i>	Velocidade	169

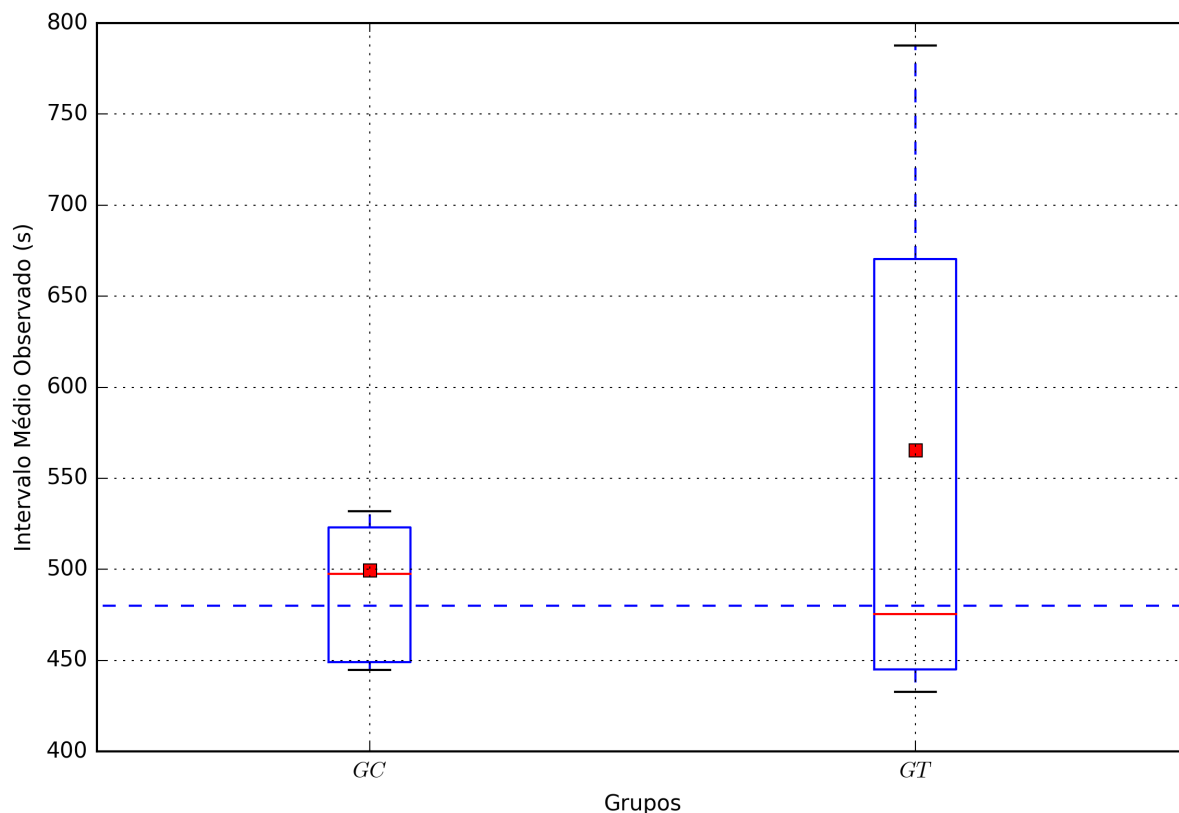
Em relação as semelhanças presentes em cada simulação, a estratégia de controle utilizada em todas foi *Forward Headway* (FH) [2], com ganho $K_c = 0.9$ e retenção máxima $r_{max} = 40$ s. Tal estratégia de controle busca normalizar os intervalos observados nas estações com base em um intervalo desejado. O intervalo desejado escolhido em todas as simulações foi de quatrocentos e oitenta (480) segundos. Esse valor é escolhido com base no tempo de ciclo observado na operação normal de um ônibus e tamanho da frota. A demanda por embarque e desembarque é constante no tempo e entre estações.

A Figura 5.1 apresenta os valores médios de intervalos observados em cada grupo. A linha horizontal contínua vermelha é o valor mediano das amostras do grupo. A base e o topo do retângulo azul indicam, respectivamente, os valores de primeiro e terceiro quartil das amostras. As linhas horizontais contínuas pretas indicam, respectivamente, o vigésimo e octogésimo percentil das amostras. O ponto vermelho representa o valor médio das amostras. A linha horizontal tracejada azul indica o valor de intervalo desejado.

A Tabela 5.2 apresenta um resumo dos resultados obtidos, sub-divididos por grupos. As variáveis usadas para estimar a qualidade dos grupos foram:

1. Intervalo Médio (*IM*), definido como a média dos intervalos, em segundos, entre ônibus observados em todas as estações;
2. Erro de Intervalo Médio (*EIM*), que expressa o erro percentual entre o Intervalo Médio e o valor de intervalo desejado;
3. Coeficiente de Variação Médio (*CVM*), que expressa a variação média percentual de cada intervalo entre ônibus observados em relação ao valor médio da simulação.

Figura 5.1: Valores médios de intervalos observados, em segundos. A linha horizontal contínua vermelha é o valor mediano das amostras. O ponto vermelho representa o valor médio das amostras. A linha horizontal tracejada azul indica o intervalo desejado. Cada caixa representa um grupo.



Fonte: Elaborada pelo autor

As expressões usadas para definir as variáveis, dado um intervalo desejado I_d , o conjunto de amostras de intervalos X , sua média \bar{X} e seu desvio padrão $\rho(X)$, foram:

$$IM = \bar{X} \quad (9)$$

$$EIM = 100 \cdot (\bar{X} - I_d) \cdot [I_d]^{-1} \quad (10)$$

$$CVM = 100 \cdot \rho(X) \cdot [\bar{X}]^{-1} \quad (11)$$

Os valores altos de CVM e EIM observados no GT são indícios de que a adaptação do controle produz resultados indesejados. Os intervalos observados nas estações ao longo da simulação são discrepantes, não normalizados e não previsíveis. Principalmente, não atingem os mesmos resultados do GC . Portanto, o objetivo de controle não é atingido.

Tabela 5.2: Resumo dos resultados obtidos, por grupo, em relação ao Intervalo Médio observado, em segundos, e ao Coeficiente de Variação dos Intervalos, em %.

Grupo	I_d (s)	IM (s)	EIM (%)	CVM (%)
GC	480	499	+04,06	$\pm 14,7$
GT	480	565	+17,8	$\pm 42,4$

A diferença percentual entre o valor médio e a mediana das amostras vista no GT (15,93%) indica que a distribuição das amostras possivelmente segue uma distribuição diferente da normal. Essa diferença não é vista no GC (0,411%). Estudos futuros podem ser realizados para identificar qual é a influência na distribuição dos dados devido ao uso do controle por velocidade.

5.1.2 Prioridade Semafórica Total

O autor do PFC, ao analisar a simulação detalhadamente, gerou a hipótese de que a discrepância entre os métodos de acionamento é criada devido às perturbações que interagem com a velocidade dos ônibus, principalmente a influência de semáforos. Portanto, para testar tal hipótese, o cenário de simulação construído no Módulo de Simulação foi adaptado para gerar prioridade semafórica total para todos os ônibus da linha modelada e reduzir a influência semafórica.

Os ônibus da linha modelada sempre encontram o semáforo em verde no seu caminho, sem levar em consideração o efeito dessa adaptação em outros elementos da malha viária, como outros veículos, passageiros ou transientes. Assim a influência das paradas de semáforos é neutralizada.

Para comprovar que a adaptação da plataforma de controle é eficiente em um ambiente sem a influência semafórica, um total de quatrocentos e oitenta e quatro (484) observações de intervalos simulados foi realizada.

Essas observações foram separadas igualmente em dois grupos de acordo as suas diferenças em relação ao método de acionamento: Grupo de Controle (GC_p), simulações com acionamento por retenção; e Grupo de Teste (GT_p), simulações com acionamento por velocidade. A Tabela 5.3 apresenta o resumo de tais grupos, contendo sua nomenclatura usada, o tipo de acionamento, e o número de observações em cada grupo.

Tabela 5.3: Características dos grupos de observações realizadas, contendo suas nomenclaturas, tipos de acionamento e número total de observações.

Grupo	Acionamento	N ^o de Observações
GC_p	Retenção	243
GT_p	Velocidade	241

Em relação as semelhanças presentes em cada simulação, a estratégia de controle utilizada em todas foi *Forward Headway* (FH) [2], com ganho $K_c = 0.9$ e retenção máxima $r_{max} = 40$ s. Tal estratégia de controle busca normalizar os intervalos observados nas estações com base em um intervalo desejado. O intervalo desejado escolhido em todas as simulações foi de quatrocentos e vinte (420) segundos. Esse valor é escolhido com base no tempo de ciclo observado na operação normal de um ônibus na linha e tamanho da frota de ônibus. A demanda de passageiros por embarque e desembarque é constante no tempo e entre estações.

Em um ambiente sem a influência semafórica, em comparação com um ambiente com a influência, a velocidade média dos ônibus aumenta. Assim, seu tempo de ciclo diminui consideravelmente. Tal redução no tempo de ciclo é o fator responsável pela redução no valor de intervalo desejado escolhido para as simulações com prioridade semafórica total.

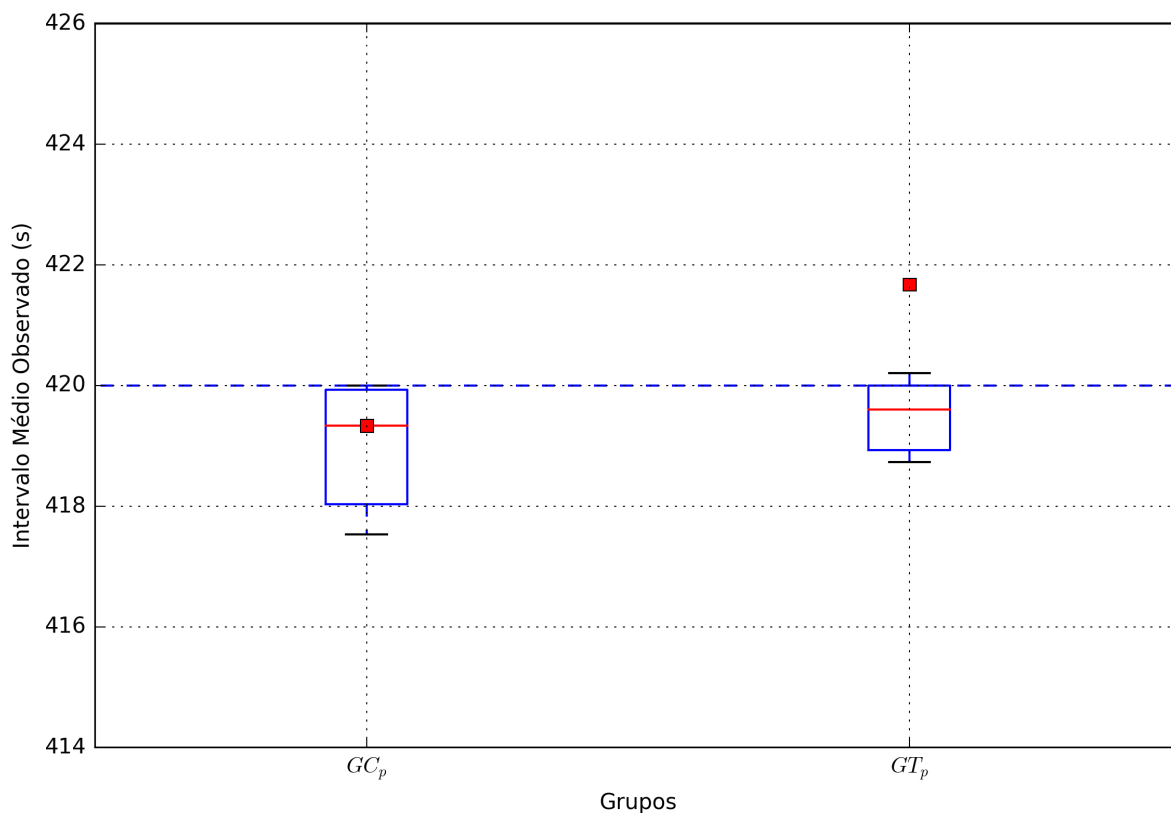
A Figura 5.2 apresenta os valores médios de intervalos observados em cada grupo, e sua apresentação segue o padrão apresentado na Figura 5.1. A Tabela 5.4 apresenta um resumo dos resultados obtidos sub-divididos por grupos, e sua apresentação segue o padrão apresentado na Tabela 5.2.

Tabela 5.4: Resumo dos resultados obtidos, por grupo, em relação ao Intervalo Médio observado, em segundos, e ao Coeficiente de Variação dos Intervalos, em %, em um ambiente sem influência semafórica.

Grupo	I_d (s)	IM (s)	EIM (%)	CVM (%)
GC_p	420	419,33	-0,15	$\pm 2,46$
GT_p	420	421,67	+0,39	$\pm 1,68$

Diferentemente do observado nas simulações onde os semáforos funcionavam normalmente, o grupo GT_p apresenta resultados satisfatórios. Ambos os métodos de acionamentos normalizam o valor médio dos intervalos observados em um valor próximo do desejado, com erros percentuais muito baixos: -0,15%, no GC_p ; e 0,39%, no GT_p .

Figura 5.2: Valores médios de intervalos observados, em simulações com prioridade semafórica, em segundos. A linha horizontal contínua vermelha é o valor mediano das amostras. O ponto vermelho representa o valor médio das amostras. A linha horizontal tracejada azul indica o intervalo desejado. Cada caixa representa um grupo.



Fonte: Elaborada pelo autor

Em relação à distribuição das amostras, em ambos os grupos a diferença percentual entre a média e a mediana é muito baixa (0,001% e 0,49% nos grupos GC_p e GT_p , respectivamente). Isso é um forte indício da distribuição seguir um padrão normal em ambos os grupos.

Tais resultados indicam que a adaptação do acionamento de controle é eficaz, realizando um sistema de controle capaz de normalizar os intervalos através da velocidade dos veículos. Contudo, para que esse efeito seja atingido, é necessária a minimização da influência dos semáforos na linha de transporte público controlada. A redução na influência semafórica realizada no projeto, através de prioridade semafórica total, não é ideal para o uso em uma aplicação real, uma vez que desconsidera seu efeito sobre outros veículos e passageiros da malha viária. Existem modos mais eficazes de atingir um sistema de prioridade semafórica, e esses são indicados para uma aplicação real.

As Figuras 5.3 e 5.4 servem de base para comparação entre os resultados obtidos com controle por velocidade entre os ambientes sem e com prioridade semafórica total. As características dos cenários sem e com prioridade são idênticas aquelas usadas nas observações, respectivamente, dos grupos GT e GT_p .

As Figuras são divididas em 3 gráficos, onde esses representam:

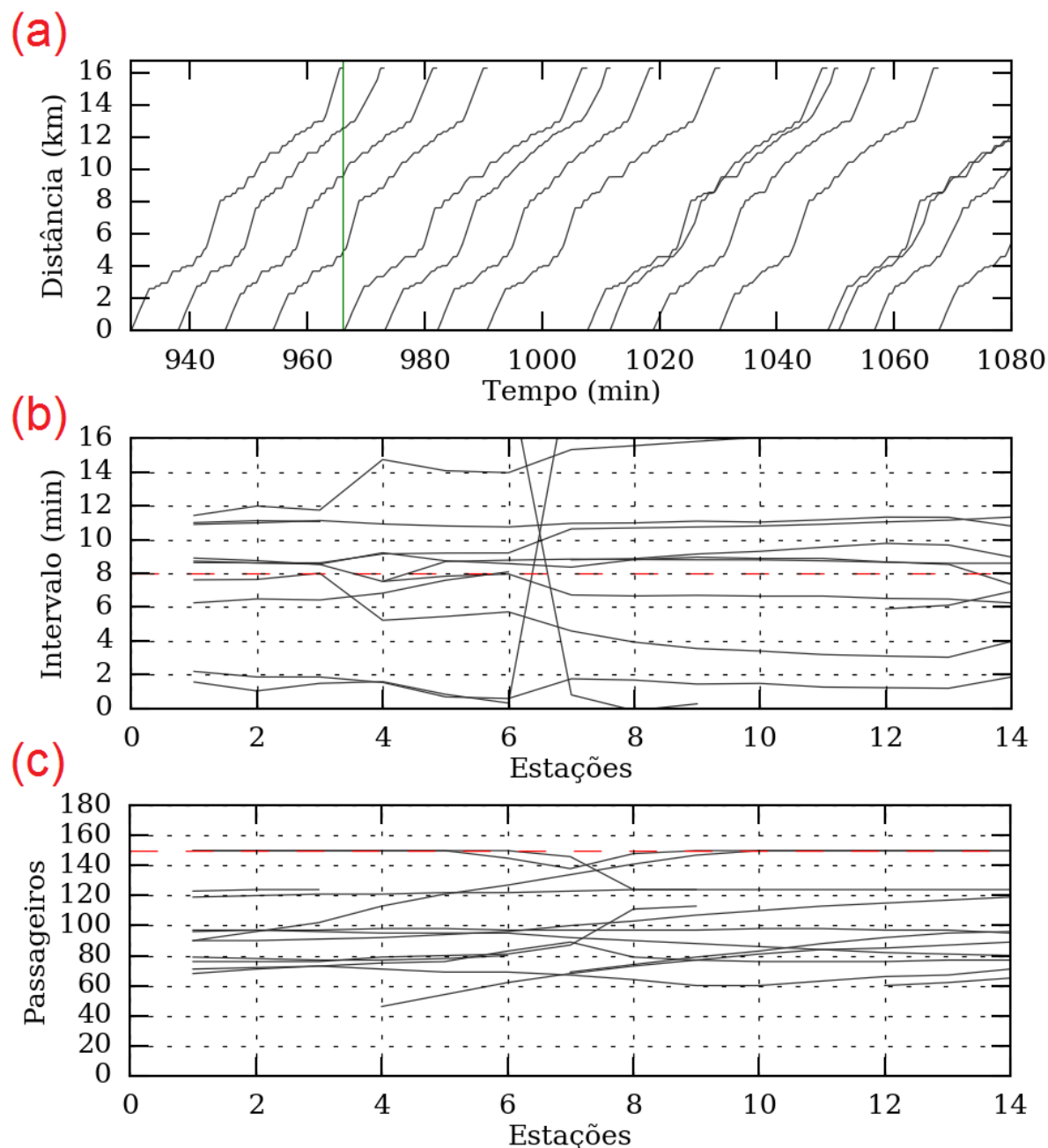
- (a). Distância percorrida ao longo do tempo. Cada linha é um ônibus da frota, e a linha verde vertical indica o instante em que o controle é iniciado;
- (b). Intervalo observado em cada estação. Cada linha é um ônibus da frota em uma determinada volta, e a linha vermelha tracejada indica o intervalo desejado;
- (c). Passageiros observados dentro do ônibus em cada estação. Cada linha é um ônibus em uma determinada volta, e a linha vermelha tracejada indica a capacidade máxima de passageiros em um ônibus.

Os resultados indicados nas figuras indicam, também, que a influência semafórica no controle por velocidade é significativa, gerando trajetórias inconstantes e não homogêneas. O intervalo observado em simulações onde é presente a influência dos semáforos é distante do ideal, em todas as quatorze estações da linha. A influência se estende, inclusive, para a distribuição dos passageiros entre ônibus: alguns ônibus trafegam com menos de 50% de ocupação; enquanto outros estão no limite de sua capacidade máxima.

Ao permitir prioridade semafórica total para os veículos da linha, suas trajetórias passam a ser reguladas como desejado. Os passageiros observados nos ônibus possuem pouca variação, mantendo aproximadamente oitenta passageiros por veículo. O intervalo entre ônibus observado nas estações é praticamente constante sobre o intervalo desejado de sete minutos.

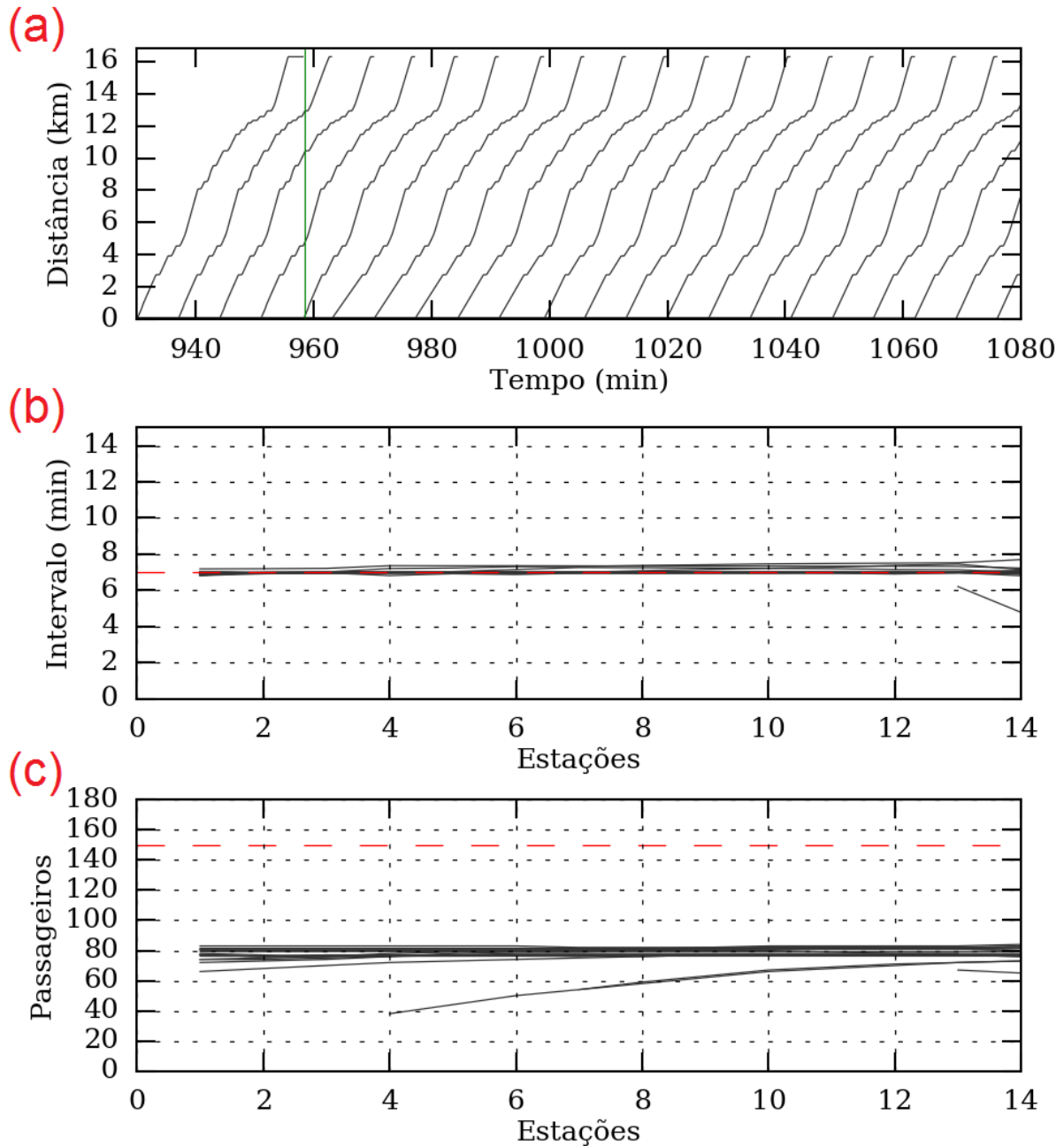
O cruzamento de linhas, visto no gráfico (b) da Figura 5.3, é devido a ultrapassagem de veículos durante a simulação. Essa característica é um reflexo do *bus bunching*, onde um dos veículos em comboiamento está realizando o processo de embarque e desembarque em uma estação quando o próximo veículo chega nessa estação. Como a simulação não permite que passageiros na fila do primeiro ônibus optem entrar no segundo ônibus, esse não realiza a parada na estação e ultrapassa o ônibus parado. Essa característica é extremamente indesejável a operação de uma linha.

Figura 5.3: Resultados gerados em simulação da malha com as características: com controle por velocidade; sem prioridade semafórica total; estratégia 'FH' com $K_c = 0.9$. Os gráficos representam: (a) distância percorrida ao longo do tempo, cada linha é um ônibus, a linha verde vertical é o momento de início do controle; (b) intervalo observado em cada estação, cada linha é um ônibus em uma volta, a linha vermelha tracejada é o intervalo desejado; e (c) número total de passageiros em cada estação, cada linha é um ônibus em uma volta, a linha vermelha tracejada é a capacidade máxima dos ônibus.



Fonte: Elaborada pelo autor

Figura 5.4: Resultados gerados em simulação da malha com as características : controle por velocidade; com prioridade semafórica total; estratégia 'FH' com $K_c = 0.9$. Os gráficos representam: (a) distância percorrida ao longo do tempo, cada linha é um ônibus, a linha verde vertical é o momento de início do controle; (b) intervalo observado em cada estação, cada linha é um ônibus em uma volta, a linha vermelha tracejada é o intervalo desejado; e (c) número total de passageiros em cada estação, cada linha é um ônibus em uma volta, a linha vermelha tracejada é a capacidade máxima dos ônibus.



Fonte: Elaborada pelo autor

5.2 Módulo de Servidores

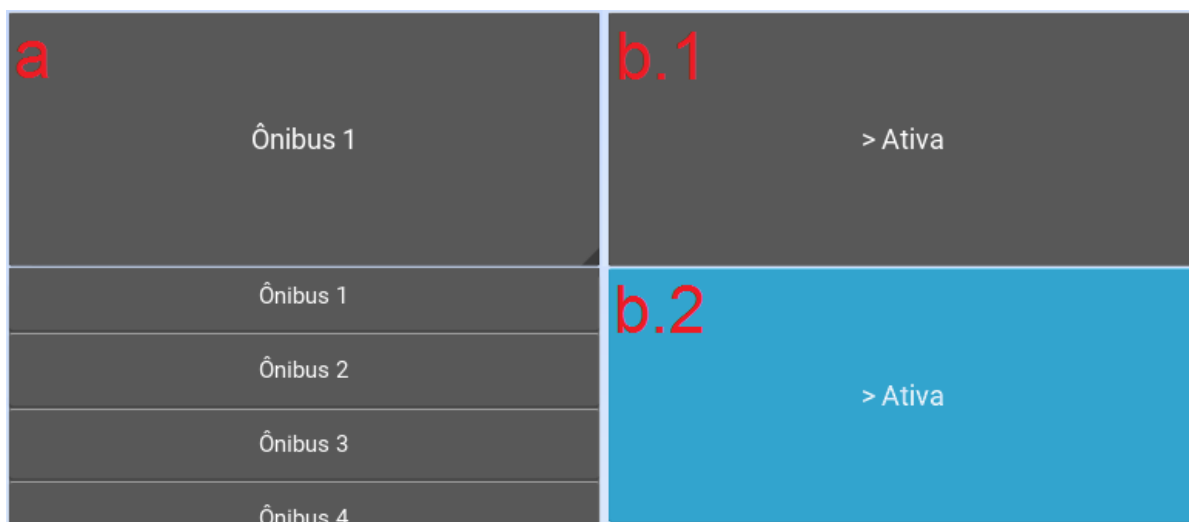
Os resultados obtidos no Módulo de Servidores serão divididos entre o aplicativo móvel desenvolvido e o sincronismo entre todos os componentes do sistema computacional de suporte.

5.2.1 Aplicativo Móvel

A fim de criar uma interface humano-maquina entre o motorista de um ônibus em uma linha controlada e o sistema de controle, um aplicativo móvel foi considerado pelo autor do PFC e seus orientadores como a melhor plataforma. Isso se deu devido à facilidade de manuseio, atual uso disseminado e ampla confiabilidade dos aparelhos portáteis atuais, como *notebooks* leves, *tablets* e celulares *smartphones*. Todos esses aparelhos podem ser facilmente conectados à Internet e inseridos ao painel de um ônibus.

De acordo com a seção 3.2.2, especificamente na Figura 3.7, o aplicativo possui dois atores principais: o Motorista, que pode iniciar o processo de controle, terminá-lo e visualizar o valor atual de controle; e o Administrador, que usa o aplicativo para informar qual é o ônibus controlado dentre a frota. A Figura 3.9 apresenta como tais operações ocorrem ao longo do tempo, e a Figura 3.10 indica as classes usadas para a implementação de tais operações.

Figura 5.5: Representação, como vistas na interface do aplicativo, das instâncias das classes (a) ‘Spinner’ e (b) ‘Button’, ambas usadas no aplicativo.



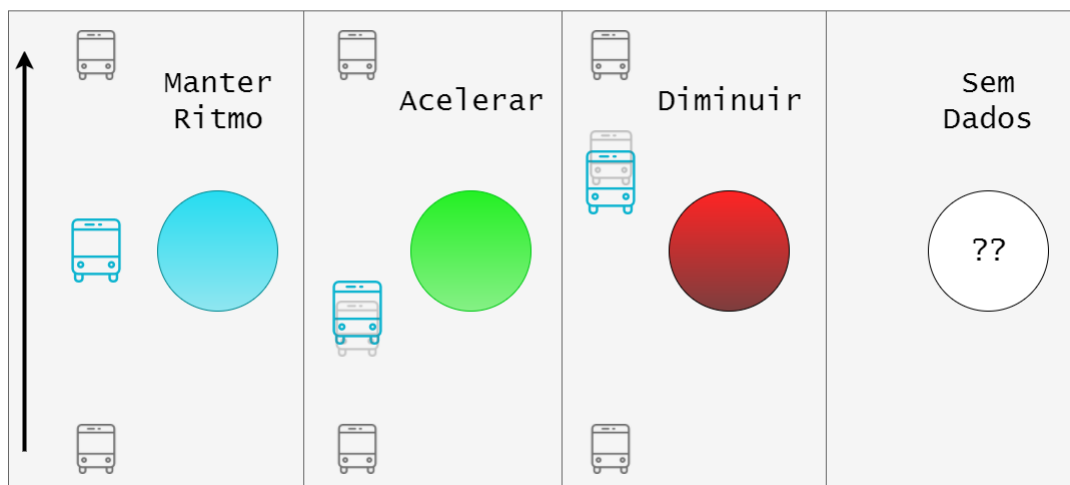
Fonte: Elaborada pelo autor

Ao criar as instâncias das classes ‘Spinner’ e ‘Button’, a interface do aplicativo as representa como apresentadas na Figura 5.5.

Ao pressionar o ‘Spinner’, um conjunto de valores é criado abaixo do botão para Escolher o Ônibus que será controlado. Tal ação é realizada pelo Administrador a linha de ônibus.

O ‘Button’ serve para Iniciar ou Terminar o Controle, sendo que ao pressionar o botão seu estado é alternado entre *normal* e *down*. A diferença visual entre tais estados está apresentada na Figura 5.5, respectivamente, em (b.1) e (b.2). Caso o botão esteja *normal*, o controle está desativado. Caso contrário, o controle está ativo. Tais ações são realizadas pelo Motorista do ônibus controlado, uma vez que devem ocorrer no momento em que o ônibus inicia ou termina sua operação.

Figura 5.6: Representação, como visto na interface do aplicativo, da instância da classe ‘Image Web’.



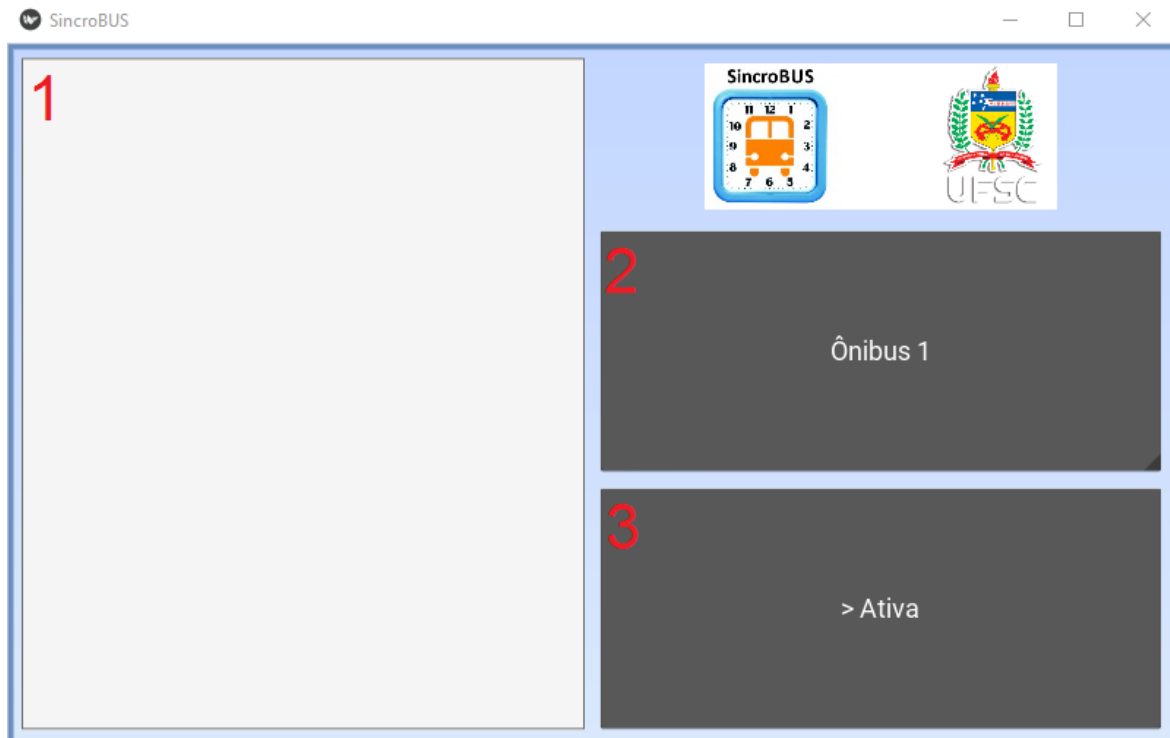
Fonte: Elaborada pelo autor

A Figura 5.6 apresenta um resumo das possibilidades de representações da instância da classe ‘Image Web’, usada para o Motorista visualizar qual é a indicação atual de velocidade. Vale a pena indicar que, no aplicativo, tais imagens são animadas, gerando uma urgência e compreensão melhor do que a imagem fixa apresentada no relatório. Existem quatro possibilidades de indicações, como especificado pelo SincroBUS: ‘Manter Ritmo’; ‘Acelerar’; ‘Diminuir’; ou ‘Sem Dados’.

A imagem ‘Manter ritmo’ é usada sempre que a velocidade do motorista foi considerada, pelo sistema de controle, como adequada para a situação. Opções ‘Acelerar’ e

‘Diminuir’ são usadas, respectivamente, quando o ônibus está atrasado ou adiantado. A opção ‘Sem Dados’ é usada sempre que o ônibus está parado em uma estação, efetuando embarque/desembarque de passageiros, ou quando a conexão com a internet é perdida, seja entre o aplicativo e o SincroBUS, ou entre o sistema de Controle e o referido servidor.

Figura 5.7: Interface gráfica do aplicativo móvel desenvolvido, em seu estado inicial. De acordo com a Figura 3.8, esse estado se chama ‘Ônibus Escolhido’, uma vez que o ‘Spinner’ já foi acionado para o Ônibus Um.



Fonte: *Print-screen* o aplicativo móvel, sistema operacional Windows 10

A Figura 5.7 representa a versão final do aplicativo, ilustrado em 3 blocos:

- (1). Representa a ‘Image Web’ que indica ao motorista qual deve ser a sua ação em função do controle atual;
- (2). Representa o ‘Spinner’ onde o Administrador a linha pode escolher qual ônibus deve ser controlado. No caso apresentado, o ônibus ‘1’ foi escolhido;
- (3). Representa o ‘Button’ onde o Motorista do ônibus pode iniciar ou terminar a operação de controle.

5.2.2 Sincronismo dos Servidores e Apresentações

Para que o sistema computacional de suporte opere de forma eficiente e atinja seus objetivos, é necessário que todos os componentes sejam capazes de enviar e receber dados de maneira simples e em tempo real. A implementação de tal capacidade foi realizada seguindo o diagrama apresentado na Figura 3.11.

Figura 5.8: Três *softwares* e suas interfaces, em tempo real, da simulação em ocorrência. (1) Representa a interface de edição e visualização do simulador Aimsun durante a simulação da linha e cenário modelados no projeto, a seta indica a posição do terceiro ônibus da frota em seu percurso. (2) É a representação, no módulo ATIS Web, do ônibus de código '3' e suas últimas informações de posição e tempo. (3) Representa a interface do aplicativo móvel desenvolvido, como seria vista pelo motorista do ônibus em questão. Todos os *print-screens* foram simultâneos.



Fonte: *Print-screens* do simulador, ATIS Web e aplicativo móvel, todos em sistema operacional Windows 10

A fim de demonstrar a capacidade de sincronismo entre os sistemas e *softwares*, projetados ou utilizados durante o PFC, a Figura 5.8 apresenta a interface de três desses durante uma simulação do cenário. Essa figura é dividida em três blocos.

O bloco '1' apresenta a interface de edição e visualização do simulador Aimsun durante a simulação da linha e cenário modelados no projeto. A seta indica a posição do terceiro ônibus da frota em seu percurso. Através da barra de controle, indicada na posição superior da imagem, o programador do *software* é capaz de controlar o fluxo temporal da simulação. A seção mais a direita de tal barra representa o instante de simulação, e na figura indica o instante '13:12:10,200'. Esse pode ser usado na criação da informação temporal dos veículos da linha.

O bloco ‘2’ representa a interface do módulo ATIS Web, do SincroBUS, durante a simulação. Nessa, um passageiro de transporte coletivo pode visualizar as informações de posição e tempo dos veículos de uma determinada linha. Na figura, o ônibus apresentado é o mesmo indicado na simulação. Sua última posição adquirida pelo servidor do SincroBUS, em formato padrão de latitude e longitude, e o instante que essa informação foi adquirida são representados em uma caixa de texto sobre o ônibus. Importante notar que o instante indicado, ‘13:12’, é o mesmo instante indicado no bloco do simulador.

O bloco ‘3’ apresenta a interface do aplicativo móvel desenvolvido durante a simulação. O ônibus escolhido para visualização do controle foi o ônibus 3, o mesmo representado nos outros dois primeiros blocos da figura. Nessa interface, o motorista do ônibus recebe a recomendação de conduta como calculada pelo controle. No caso da figura, essa recomendação é de acelerar o veículo.

6 Conclusão e Discussão

O projeto visa a implementação de um sistema computacional de suporte à operação controlada de trânsito coletivo urbano. Busca-se orientar motoristas de ônibus no sentido da regularização dos intervalos entre veículos da mesma linha, através de um controle de velocidade. Esse controle é necessário, uma vez que os ônibus têm dificuldade de manter o plano de horários ou intervalo planejado.

Durante o curso do projeto, foi desenvolvido um cenário de simulação da malha viária como prevista para Florianópolis, assunto de demasiada importância para a região. As modificações previstas na região metropolitana da cidade são vigentes e atuais, indicando a necessidade de estudos sobre o tráfego urbano para melhorias eficazes.

O sistema computacional criado simulou uma estrutura que poderá ser vista em sistemas de tráfego inteligentes, com armazenamento, organização e distribuição da informação de maneira sucinta e eficiente. A exportação de dados em formatos padrão, como aqueles necessários para a operação do SincroBUS ou Google Maps Transit, é essencial nesse objetivo, uma vez que permite que todos os atores de importância em uma malha viária possam se comunicar com clareza e sem ambiguidade. As ferramentas utilizadas para construir o formato padrão nos dados de posição foram eficientes em muitos casos, mas com erros visíveis em algumas instâncias, e deverão ser estudadas e aperfeiçoadas ao implementar um sistema de suporte em uma aplicação real. Estudos futuros devem ser realizados para a implementação de armazenamento em banco de dados relacional, uma vez que esse ainda é muito usado em aplicações reais.

O aplicativo para dispositivos móveis construído apresentou resultados consistentes com o objetivo inicial, com indicações de conduta ao motorista simples, claras e eficazes. Tal aplicativo, mesmo em sua estrutura simples, pode ser usado no acionamento de controle por velocidade sem problemas maiores, dado um servidor compatível. Prevê-se que o aplicativo móvel possa ser modificado para melhorar suas capacidades, como: apresentar e armazenar mais informações importantes da linha controlada; servir de sensor veicular, captando informações como posição, velocidade, aceleração, consumo energético, entre outras; possibilitar a comunicação direta do motorista com a central de controle em casos de acidentes ou problemas na operação.

Por outro lado, o aplicativo desenvolvido não pode ser exportado para um dispositivo móvel, com sistema operacional Android, como previsto. Isso se deu devido a erros desconhecidos pelo autor e ao pouco tempo disponível para o estudo da biblioteca Kivy, usada no desenvolvimento do aplicativo. Não obstante, o aplicativo funciona perfeitamente no sistema operacional Windows 10, onde foi desenvolvido. O autor e seus orientadores consideraram esse resultado satisfatório, uma vez que o aplicativo funciona perfeitamente, e a portabilidade para o sistema operacional originalmente previsto é apenas técnico.

O controle operacional via regularização de velocidade de veículos construído no projeto apresentou resultados satisfatórios, como previsibilidade e estabilidade, quando utilizado em conjunto com a prioridade semafórica total. Por outro lado, a influência semafórica de uma malha onde não há prioridade foi depreciativa ao controle, desestabilizando o intervalo observado na simulação. As vantagens obtidas pelo controle por velocidade, como a melhor aceitação dos passageiros, só podem ser atingidas se um sistema de prioridade semafórica for também implementado. Para esse fim, o sistema computacional de suporte apresenta características desejadas, como armazenamento e organização de dados relativos à prioridade. Não obstante, o sistema de prioridade apresentado no projeto não é eficiente do ponto de vista global da malha viária, uma vez que desconsidera a sua influência sobre outros veículos e passageiros da malha, como carros e pedestres. Estudos futuros devem ser realizados para a integração de prioridade semafórica otimizada em sistemas que deseja-se implementar controle de intervalos via regularização de velocidade.

Bibliografia

- [1] G. F. Newell and R. B. Potts. Maintaining a bus schedule. In *Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne*, volume 2, 1964.
- [2] L. Zimmermann. Controle de intervalos entre Ônibus: Comparação entre métodos realimentados e controle preditivo. page 84, 2016.
- [3] O. J. Ibarra-Rojas, F. Delgado, R. Giesen, and J. C. Muñoz. Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological*, 77:38–75, 2015.
- [4] J. Argote-Cabanero, C. F. Daganzo, and J. W. Lynn. Dynamic control of complex transit systems. *Transportation Research Part B: Methodological*, 81:146–160, 2015.
- [5] Andre Carrel, Rabi G Mishalani, Raja Sengupta, and Joan L Walker. In pursuit of the happy transit rider: dissecting satisfaction using daily surveys and tracking data. *Journal of Intelligent Transportation Systems*, 20(4):345–362, 2016.
- [6] Lepidus Tecnologia LTDA ME. Manual - software supervisorio para integração dos componentes do sistema experimental de rastreamento e controle de veículos de transporte público, 2009.
- [7] GanttProject. Free project scheduling and management software v. 2.8.5, disponível em <<http://www.ganttproject.biz/>>, Acesso em 21/06 de 2017.
- [8] Python Software Foundation. Disponível em <<https://www.python.org/psf/>>. Acesso em 04/02 de 2017.
- [9] TIOBE Software BV. Disponível em <<http://www.tiobe.com/tiobe-index/>>. Acesso em 03/02 de 2017.
- [10] Kivy Community. Disponível em <<https://kivy.org>>. Acesso em 15/05 de 2017.
- [11] Kivy Community. Disponível em <<https://kivy.org/docs/guide/lang.html>>. Acesso em 15/05 de 2017.
- [12] TSS. Aimsun dynamic simulator users manual v. 7. 2012.

- [13] S. Prolo. Reformulação de uma plataforma de microssimulação do sistema de controle de intervalos de ônibus em corredores exclusivos. pages 16–18, 2016.
- [14] Google Developers. Disponível em <<https://developers.google.com/transit/gtfs>>. Acesso em 25/05 de 2017.
- [15] Object Management Group. Disponível em <<http://www.omg.org/spec/uml/2.5/pdf/>>. Acesso em 30/05 de 2017.
- [16] International Organization for Standardization. Disponível em <<https://www.iso.org/standard/32620.html>>. Acesso em 30/05 de 2017.
- [17] Logit Engenharia Consultiva, Strategy&, and Machado Meyer Sendacz Opice Advogados. *Plano de Mobilidade Urbana Sustentável da Grande Florianópolis - PLAMUS*. Florianópolis, 2015.
- [18] PLAMUS. Plano de mobilidade urbana sustentável da grande florianópolis, histórico para imprensa, disponível em <<http://www.plamus.com.br/imprensa.html>>, Acesso em 07/07 de 2017.
- [19] Rosetta Code Community. Disponível em <https://rosettacode.org/wiki/haversine_formula>. Acesso em 11/05 de 2017.
- [20] International Organization for Standardization. Disponível em <<https://www.iso.org/iso-8601-date-and-time-format.html>>. Acesso em 11/05 de 2017.