

Semafácil: Sistema para controle semafórico remoto baseado na Internet das Coisas

Pedro J. Gil¹, Rodolfo F. Oliveira²

¹Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Hortolândia (IFSP)
Av. Thereza Ana Cecon Breda, 1896 - Vila Sao Pedro, Hortolândia - SP, 13183-250

juan.franco@aluno.ifsp.edu.br, rodolfo.oliveira@ifsp.edu.br

Abstract. *The traffic light control system is crucial for the organization and safety of urban traffic but faces challenges that are necessary for modernization. This project proposes the development of a remote traffic light control system based on the Internet of Things (IoT). With a prototype of a fixed-time traffic light control, the solution integrates a web interface, PHP backend, Python backend, and a database on low-cost hardware. The conclusion highlights the overcome challenge, showcasing the potential of IoT in the modernization of traffic light control systems.*

Resumo. *O sistema de controle semafórico é crucial para a organização e segurança do trânsito urbano, mas enfrenta desafios necessários para a modernização. Este projeto propõe o desenvolvimento um sistema remoto de controle semafórico baseado na Internet das Coisas (IoT). Com um protótipo de controle semafórico de tempo fixo, a solução integra uma interface web, backend PHP, backend Python e banco de dados em um hardware de baixo custo. A conclusão destaca o desafio superado, evidenciando o potencial da IoT na modernização dos sistemas de controle semafórico.*

1. Introdução

No contexto do trânsito urbano, o sistema de controle semafórico desempenha um papel fundamental na organização e segurança dos veículos e pedestres. Os controladores semafóricos são responsáveis por administrar a duração das fases dos semáforos, permitindo a fluidez adequada do tráfego em cruzamentos e vias urbanas. No entanto, esses controladores apresentam problemas e deficiências que podem ser solucionados por meio de modernização e aprimoramento do sistema.

Problemas como falta de sincronização entre semáforos, falhas no funcionamento e dificuldades na configuração e manutenção são frequentes. Diante desses problemas, há a necessidade de desenvolver um sistema semafórico eficiente, acessível e de fácil manutenção, que seja capaz de melhorar o fluxo do tráfego urbano e minimizar congestionamentos.

Portanto o objetivo deste trabalho é desenvolver um sistema semafórico de controle remoto baseado em uma interface *web*. O sistema proposto terá como foco principal a implementação de um controlador semafórico que permitirá o gerenciamento do tempo e estado dos semáforos.

O protótipo funcional será composto por um *hardware* de baixo custo para o gerenciamento local dos semáforos. Além disso, a interface *web* permitirá o controle remoto e a configuração dos semáforos, com o intuito de propor maior agilidade e facilidade de uso para o operador semafórico.

O trabalho está dividido em 8 seções: Revisão Bibliográfica, abordando temas como "Internet das Coisas" e "Controlador Semafórico"; Trabalhos Correlatos, onde os trabalhos semelhantes a este encontrados na literatura são apresentados; Metodologia; Proposta do Trabalho; Materiais Utilizados; Desenvolvimento e Conclusão.

2. Revisão bibliográfica

Este capítulo apresenta uma revisão bibliográfica abrangendo três assuntos relevantes para o desenvolvimento do projeto Semafácil: Internet das Coisas (*IoT*), Controlador Semafórico e Computador de Placa Única ou *Single-Board Computer*. A interseção desses três temas tem um impacto significativo na implementação do sistema.

2.1. Internet das Coisas

A Internet das Coisas, ou abreviadamente *IoT* do inglês "*Internet of Things*", trata-se de um paradigma tecnológico que se baseia na interconexão de dispositivos físicos pelo mundo utilizando a Internet. De acordo com [Gokhale et al. 2018], "A Internet das Coisas é a internet dos objetos físicos - dispositivos, instrumentos, veículos, construções embarcados com eletrônicos, circuitos, *software*, sensores e conectividade - que habilita esses objetos para coletar e trocar informações". O conceito de *IoT* também está intrinsecamente ligado a sistemas eficientes e de pequena escala, conforme mencionado em [IEEE 2015], "Os sistemas operacionais de *IoT* são projetados para serem executados em componentes de pequena escala da maneira mais eficiente possível, ao mesmo tempo em que fornecem funcionalidades básicas para simplificar e suportar o sistema global de *IoT* em seus objetivos e propósitos".

A definição de *IoT* como conceito foi elaborada pelo pioneiro britânico em tecnologia chamado Kevin Ashton, em 1999. Segundo [Ashton 1999], "(...) Se tivéssemos computadores que soubessem tudo o que há para saber sobre as coisas - usando dados que eles coletaram sem qualquer ajuda nossa - seríamos capazes de rastrear e contar tudo, reduzindo enormemente o desperdício, as perdas e os custos".

Em suma, a *IoT* permite a aproximação da eficiência, complexidade e autonomia tecnológica com o mundo presente, gerando informações em tempo real através de sensores. Além disso, ela possibilita que os seres humanos se conectem de maneira mais profunda e significativa com o ambiente ao seu redor, possibilitando interações inteligentes e informadas que impulsionam a inovação, a tomada de decisões mais assertivas e a melhoria contínua das experiências humanas.

2.2. Controlador Semafórico

Um controlador semafórico é responsável por gerir semáforos, conforme citado por [CONTRAN 2014]¹: "São os equipamentos programáveis que comandam as trocas das

¹CONTRAN: Conselho Nacional de Trânsito: é o órgão máximo normativo e consultivo do Sistema Nacional de Trânsito. Ele elabora diretrizes da Política Nacional de Trânsito e coordena todos os órgãos do Sistema Nacional de Trânsito. [dos Transportes 2023]

indicações luminosas dos grupos focais”, os quais são elementos essenciais para o controle de tráfego nas vias urbanas, regulando o fluxo de veículos e pedestres em cruzamentos e interseções. Nesta seção, será apresentada uma revisão bibliográfica sobre controladores semafóricos, abordando seus conceitos, tecnologias e estudos relacionados.

De acordo com [CONTRAN 2014], um controlador semafórico é dividido em dois tipos: Controladores Eletromecânicos, cuja programação é implementada por meio de recursos mecânicos limitados exclusivamente ao *hardware*, e os Controladores Eletrônicos, cuja programação é realizada por componentes com capacidade computacional, permitindo a adoção de diferentes estratégias de controle de tráfego.

Os principais elementos de um controlador semafórico, conforme [CONTRAN 2014], são: plano semafórico, estágio e diagrama de intervalo luminoso. Um plano semafórico é “um conjunto de elementos que caracteriza a programação da sinalização semafórica para uma interseção ou seção de via, em um determinado período do dia”. Um estágio é um intervalo de tempo em que um ou mais grupos de movimentos recebem simultaneamente o direito de passagem. Por sua vez, um diagrama de intervalo luminoso consiste na representação da duração e sequência dos intervalos luminosos e estágios por meio de barras horizontais, associando-os aos grupos semafóricos correspondentes, onde cada estágio possui suas cores ativas, ou intervalos luminosos, período de tempo em que permanece inalterada a configuração luminosa dos semáforos (grupos focais) que controlam o tráfego em um determinado local, em um intervalo de segundos total, representando um ciclo semafórico.

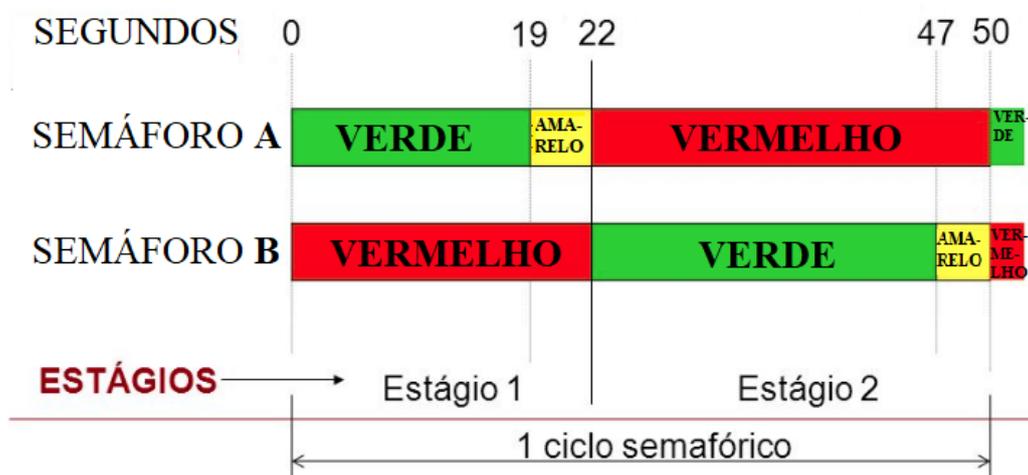


Figura 1. Diagrama de intervalo luminoso
Fonte: Elaborado pelo próprio autor - 2023

Na figura 1 podemos observar um diagrama de intervalo luminoso que determina o funcionamento de dois semáforos, especificando as cores e a duração de sua ativação. Na Figura, cada estágio é associado a suas respectivas cores. Esse intervalo, expresso em segundos totais, representa um ciclo semafórico.

A eficiência do controle de tráfego nas áreas urbanas está diretamente relacionada ao funcionamento adequado dos controladores semafóricos, os quais desempenham um papel crucial no controle do tráfego urbano. Esses dispositivos asseguram a fluidez e a

segurança nas vias, contribuindo para a organização e a otimização do tráfego veicular e de pedestres.

2.3. Computador de Placa Única - Single-Board Computer

Um Computador de Placa Única, do inglês, *Single-Board Computer* (SBC), é um dispositivo que oferece uma infraestrutura de *hardware* pequena porém robusta, facilitando o desenvolvimento de *software* para a *IoT*. Um SBC é composto por uma única placa de circuito integrado, contendo processador, armazenamento e memória. A maioria dos SBCs possui interfaces de entrada e saída, permitindo a conexão de diversos tipos de sensores. De acordo com [Kanagachidambaresan 2021], "Os Single Board Computers normalmente são projetados para ter todas as opções de processamento e comunicação em uma única placa de circuito impresso, principalmente fabricada para atender ao propósito de baixo consumo de energia e portabilidade". Conforme mencionado por [Isikdag 2015], "No futuro, a Internet não será apenas um meio de comunicação para as pessoas; na verdade, será um ambiente de comunicação para dispositivos."

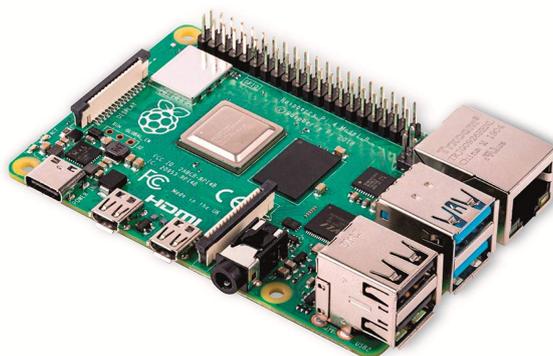


Figura 2. Exemplo de Single-board computer - Raspberry Pi
Fonte: <https://www.raspberrypi.com/> - 2023

3. Trabalhos correlatos

Esta seção apresenta os trabalhos encontrados na literatura que se assemelham ao trabalho presente. Após definir a motivação do trabalho, foram selecionados os principais termos de pesquisa: *IoT*, luzes de tráfego, semáforos, controle semafórico, *dashboard*, *Raspberry Pi* e *Arduino*. A pesquisa foi feita utilizando os termos também traduzidos para a língua inglesa. A principal plataforma escolhida para a pesquisa dos trabalhos correlatos foi o *Google Scholar*.

Após realizar as buscas, foram encontrados diversos trabalhos correlatos, abordando a temática do controle semafórico sob diferentes aspectos. No entanto, a fim de garantir a relevância e qualidade dos estudos selecionados, estabeleceram-se critérios específicos:

Uso de tecnologias similares: Foram considerados apenas os artigos que estivessem diretamente relacionados à criação de soluções para controle semafórico utilizando tecnologias relacionadas à *IoT*, com uma interface gráfica ou *web* para controle remoto.

Relevância tecnológica: Foram selecionados os trabalhos correlatos que abordassem o tema do controle semafórico, contribuindo para a contextualização dos problemas e tecnologias relacionadas.

Com base nesses critérios, os seguintes trabalhos foram escolhidos:

”*Development of a Network-Enabled Traffic Light System*”² de [Onn et al. 2014]: Este trabalho consiste no desenvolvimento de um protótipo de controle semafórico que utiliza um microcomputador para enviar comandos às luzes semafóricas e um *smartphone* para enviar informações ao microcomputador. O usuário pode realizar mudanças e controlar remotamente o comportamento dos semáforos por meio de uma interface gráfica criada utilizando o *Python Tkinter*³. Esse sistema permite uma redução dos custos de manutenção a longo prazo, sendo o controle de tráfego efetuado utilizando a lógica *fuzzy*⁴.

”*Controle experimental de semáforos vía web con Raspberry Pi y PiTraffic*” de [Otero 2022]: Este trabalho aborda o controle semafórico de forma similar ao trabalho anterior, porém com algumas diferenças cruciais e adições relevantes. O *Raspberry Pi* é utilizado como um servidor *web*, hospedando uma página acessível por qualquer dispositivo cliente. Isso permite que as lâmpadas do semáforo sejam controladas a partir de qualquer navegador, em vez de ser restrito a uma interface específica.

”*Web Service-based Automation System for Duration Scheduling and Remote Control of Traffic Signal Lights*”⁵ de [Dandil and Gültekin 2017] aborda o crescente problema de tráfego nas cidades e a necessidade de reorganizar os sistemas de sinalização de tráfego. O autor da pesquisa propõe uma alternativa econômica para o gerenciamento do controle de semáforos, possibilitando a alteração da duração das luzes remotamente pela Internet. O sistema proposto utiliza um *Arduino*, que pode ser programado remotamente por meio de um serviço *web*. Vale destacar que não é necessário possuir conhecimento em programação para gerir o sistema, uma vez que este é direcionado especialmente para os agentes de trânsito.

Após a análise dos trabalhos relacionados, elaborou-se a Tabela 1, que condensa algumas das características principais de cada sistema, simplificando a comparação entre eles e o presente trabalho.

O trabalho ”Semafácil: Sistema para Controle Semafórico Remoto” tem como objetivo o desenvolvimento e prototipagem de um sistema de controle semafórico baseado em uma interface *web* de baixo custo. Para isso, utiliza-se um *single-board computer* como controlador dos semáforos, hospedando uma página *web* pela qual as configurações são realizadas, acessível a partir de qualquer navegador. Ao contrário de trabalhos correlatos, este projeto concentra-se principalmente na incorporação de funcionalidades que se

²*Development of a Network-Enabled Traffic Light System* ou Desenvolvimento de um Sistema de Semáforo com Conectividade de Rede

³O pacote *tkinter* (“*Tk interface*”) é a interface padrão do *Python* para a biblioteca de ferramentas gráficas [Python¹ 2023]

⁴A lógica *fuzzy*, ou por vezes chamada, lógica nebulosa, surge através de sua formulação da teoria dos conjuntos para gerar um certo ”afrouxamento” da rigidez numérica da matemática clássica. [BARROS 2019]

⁵*Web Service-based Automation System for Duration Scheduling and Remote Control of Traffic Signal Lights* ou Sistema de Automação Baseado em Serviços Web para Agendamento de Duração e Controle Remoto de Sinalizações de Semáforos de Tráfego

Trabalhos	Controle remoto do sistema semafórico	<i>Uso de webservice</i>	Controle manual das luzes semafóricas	Configuração por intervalo luminoso	Status dos semáforos em tempo real	Configuração por horários e dias da semana
Semafácil	X	X		X	X	X
Controle experimental de semáforos via con Raspberry Pi y PiTraffic	X	X	X		X	
Web Service-based Automation System for Duration Scheduling and Remote Traffic Signal Lights	X		X			
Development of a Network-Enabled Traffic Light System	X		X			

Tabela 1. Comparação entre Semafácil e seus trabalhos correlatos

baseiam em horários e dias da semana para o controle semafórico.

4. Metodologia

A metodologia adotada neste trabalho de desenvolvimento de controle semafórico é a incremental iterativa. O método incremental iterativo é uma abordagem de desenvolvimento de software que envolve ciclos repetitivos de entrega incremental. Conforme descrito por [SOMMERVILLE 2016], "O desenvolvimento incremental se baseia na ideia de desenvolver uma implementação inicial, obter *feedback* dos usuários ou terceiros e fazer o *software* evoluir por meio de várias versões, até alcançar o sistema necessário", com funcionalidades refinadas no processo iterativo ao longo dos ciclos.

Os principais ciclos de um projeto incluem fases de análise, projeto, implementação e testes. Nas fases de análise, é realizado o levantamento dos requisitos do sistema, abordando as necessidades e expectativas relevantes do projeto, transformando-os em requisitos funcionais. Na etapa de projeto, são definidas as soluções técnicas para atender aos requisitos levantados, envolvendo a criação de diagramas e documentação detalhada do software.

Na fase de implementação, ocorre a escrita efetiva do código-fonte, atendendo aos requisitos pré-definidos. Além disso, são realizados diversos testes para verificar as funcionalidades, desempenho e qualidade do projeto.

A escolha dessa metodologia se deve às suas diversas vantagens para o desenvolvimento do sistema, como a facilidade em atualizar e refinar os requisitos ao longo do processo de desenvolvimento e a agilidade na execução do projeto. Ela permite uma abordagem iterativa e incremental, possibilitando ajustes e melhorias contínuas, acompanhando as mudanças e demandas do projeto.

5. Proposta

A proposta do projeto é desenvolver um protótipo de um sistema de controlador semafórico de tempo fixo, utilizando uma interface *web*. Como se trata de um protótipo, o gerenciamento do sistema foi focado em semáforos duplos de três fases (verde, amarelo e vermelho). O sistema será composto por diferentes componentes: uma interface

de usuário *web front-end*, destinada à configuração do controlador semafórico; um *back-end* em PHP responsável por manipular cadastros, edições e operações relacionadas à configuração dos semáforos, além de controlar um processo em *Python*, ligando e desligando o mesmo; um *back-end* em *Python* responsável por executar as instruções cadastradas no banco de dados e informar em tempo real o status dos semáforos; e um banco de dados contendo informações relevantes sobre o sistema e as luzes semafóricas. Todos esses componentes estão abrigados em um *single-board computer* de baixo custo.

O sistema, conforme ilustrado na Figura 3, opera da seguinte maneira: o sistema Semafácil, hospedado em um *single-board computer*, está conectado à internet por meio de um chip de internet sem fio e é responsável por trocar informações e enviar comandos para os semáforos, hospedando o sistema web denominado "Semafácil". O usuário conecta-se ao sistema *web* hospedado e pode visualizar e enviar comandos para o *single-board computer*, que controla o funcionamento dos semáforos.

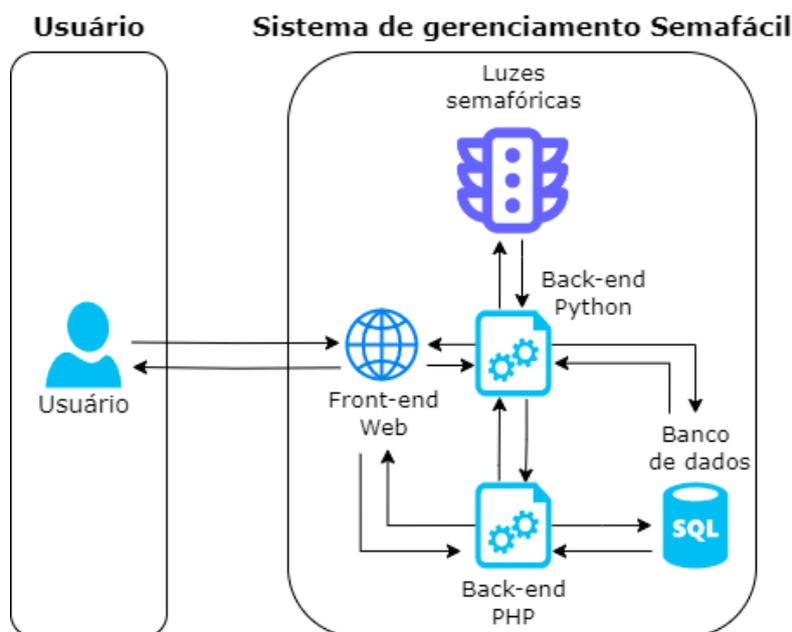


Figura 3. Diagrama arquitetural Semafácil
Fonte: Elaborado pelo próprio autor - 2023

5.1. Requisitos

De acordo com [SOMMERVILLE 2016], "a Engenharia de Requisitos normalmente é apresentada como o primeiro estágio do processo de engenharia de *software*". Portanto, o levantamento de requisitos é fundamental para definir o escopo do sistema. Todo desenvolvimento de *software* é constituído por dois principais tipos de requisitos: os funcionais e os não funcionais. "Os requisitos funcionais de um sistema descrevem o que ele deve fazer e dependem do tipo de software que está sendo desenvolvido"[SOMMERVILLE 2016]. Nos tópicos a seguir, serão apresentados os requisitos funcionais do presente sistema. Devido ao escopo e tempo, este trabalho não abordou o levantamento e consequentemente desenvolvimento dos requisitos não funcionais.

5.2. Requisitos funcionais

Os requisitos funcionais do presente sistema, destacados na Tabela 2, incluem um *CRUD*, (do inglês *Create, Read, Update, Delete*) de "Perfil de Intervalo Luminoso", que representa o perfil de um diagrama luminoso definindo o acendimento das luzes semafóricas; "Intervalo de Tempo", que define o horário de funcionamento dos semáforos; "Intervalo de Dias", que define os dias de funcionamento dos semáforos; um programa de integração dos semáforos com as informações do banco de dados, responsável por realizar a integração entre o banco de dados e os semáforos executando as configurações registradas pelos *CRUDs* anteriores; "Exibir status dos semáforos", que mostra em tempo real o acendimento das luzes semafóricas; e "Ligar e desligar o sistema semafórico remotamente", permitindo o controle do programa de integração dos semáforos.

Conforme [MOZILLA 2023], *CRUD* (*Create, Read, Update, Delete*) é um acrônimo para as maneiras de se operar em informação armazenada, envolvendo as quatro operações básicas de armazenamento persistente. A representação dos principais requisitos funcionais do sistema é:

RF01: Gerenciamento do Perfil de Intervalo Luminoso. Descrição: O sistema deve permitir a criação, leitura, atualização e exclusão de Perfis de Intervalo Luminoso, semelhantes a "Planos semafóricos" conforme definido por [CONTRAN 2014] como "o conjunto de elementos que caracteriza a programação da sinalização semafórica para uma interseção ou seção de via, num determinado período do dia". Esses planos são responsáveis por determinar o funcionamento dos semáforos A e B. Cada perfil inclui as seguintes informações: duração do acendimento das luzes verde, amarelo e vermelho para os semáforos A e B, uma descrição do perfil e a indicação do funcionamento do amarelo intermitente. O amarelo intermitente, definido por [CONTRAN 2014] como aquele que "adverte da existência de situação perigosa ou obstáculo". O gerenciamento do perfil de intervalo luminoso é crucial para assegurar o correto funcionamento do sistema de controle semafórico.

RF02: Gerenciamento do Intervalo de Tempo O sistema deve permitir a criação, leitura, atualização e exclusão de Intervalos de Tempo, os quais definem os horários de funcionamento dos semáforos. Cada intervalo possui um horário de início e um horário de fim. Esses intervalos estão diretamente relacionados aos Perfis de Intervalo Luminoso, determinando o tempo de funcionamento dos semáforos A e B em cada período do dia. O gerenciamento do intervalo de tempo é crucial, pois permite a configuração adequada dos horários de operação do sistema de controle semafórico.

RF03: Gerenciamento do Intervalo de Dias O sistema deve permitir a criação, leitura, atualização e exclusão de Intervalos de Dias, os quais representam os dias da semana. Cada intervalo de dias está relacionado aos Intervalos de Tempo e define em quais dias da semana esses intervalos de tempo são aplicados. Por exemplo, é possível configurar diferentes intervalos de tempo para os dias úteis e fins de semana. O gerenciamento do intervalo de dias viabiliza a configuração dos dias nos quais os Intervalos de Tempo serão aplicados no sistema de controle semafórico.

RF04: Programa de Integração dos Semáforos com as Informações do Banco de Dados As informações cadastradas nos *CRUDs* são executadas dinamicamente. O programa realiza uma leitura inicial das programações semafóricas e inicia o monitora-

mento do tempo atual em horas e minutos. Quando o programa detecta que o tempo atual pertence a uma programação, inicia a execução intermitente da programação até que ela seja concluída. Durante esse intervalo, o programa continua verificando se há outras programações.

RF05: Exibir Status dos Semáforos em Tempo Real O sistema deve fornecer uma funcionalidade para exibir em tempo real o status de funcionamento dos semáforos. Isso permitirá que o controlador semafórico observe e monitore o funcionamento dos semáforos A e B. Através dessa funcionalidade, o controlador poderá identificar possíveis falhas ou problemas de funcionamento, garantindo uma manutenção eficiente do sistema. O status dos semáforos inclui informações como o tempo restante em cada fase (vermelho, verde, amarelo) e poderá fornecer indicadores visuais ou alertas em caso de falhas ou anomalias. A exibição do status dos semáforos em tempo real auxilia na gestão e no monitoramento adequado do sistema de controle semafórico.

RF06: Ligar e Desligar o Sistema Semafórico Remotamente Este requisito possibilita que o usuário tenha controle total sobre os serviços semafóricos, incluindo tanto a exibição do status dos semáforos em tempo real quanto o programa de integração dos semáforos com as informações do banco de dados.

Tabela 2. Requisitos funcionais

ID	Iteração	Requisito	Prioridade	Descrição Detalhada
RF01	Iteração 1	Gerenciamento do Perfil de Intervalo Luminoso	Média	Cadastrar, editar e remover diagramas de intervalo luminoso.
RF02	Iteração 2	Gerenciamento do Intervalo de Tempo	Média	Cadastrar, editar e remover os intervalos de tempo no qual o intervalo luminoso é aplicado.
RF03	Iteração 3	Gerenciamento do Intervalo de Dias	Média	Cadastrar, editar e remover os dias nos quais os intervalos de tempo são aplicados.
RF04	Iteração 4	Programa de integração dos semáforos com as informações do banco de dados	Alta	Integração entre o banco de dados e os semáforos executando as configurações previamente salvas.
RF05	Iteração 5	Exibir Status dos Semáforos em Tempo Real	Baixa	Exibir o status das luzes semafóricas, se estão ligadas ou desligadas.
RF06	Iteração 5	Ligar e desligar o sistema semafórico remotamente	Baixa	Controle do funcionamento do programa de integração dos semáforos.

6. Diagrama de caso de uso

O diagrama de caso de uso apresentado na Figura 4 ilustra as interações entre o ator e o sistema de controle semafórico. O ator principal é o "Controlador Semafórico", que

desempenha um papel central nas operações do sistema e se comunica com a entidade chamada "Central Semafórica".

Os casos de uso identificados são os seguintes:

1. **Acessar *CRUDs* de Intervalo de Tempo:** O Controlador Semafórico tem a capacidade de acessar as funcionalidades relacionadas ao intervalo de tempo do sistema. Isso inclui a criação, remoção, atualização e exclusão de informações sobre os horários de funcionamento dos semáforos.

2. **Acessar *CRUDs* de Perfil de Intervalo Luminoso:** O Controlador Semafórico pode interagir com as funcionalidades relacionadas ao perfil de intervalo luminoso. Isso envolve a realização de operações CRUD para definir e modificar os tempos de duração das luzes verde, amarela e vermelha de cada semáforo, bem como a configuração do amarelo intermitente e a descrição do perfil.

3. **Acessar *CRUDs* de Intervalo de Dias:** O Controlador Semafórico tem a capacidade de acessar as funcionalidades relacionadas ao intervalo de dias do sistema. Isso permite definir quais dias da semana estão associados a determinados intervalos de tempo e perfis de intervalo luminoso, possibilitando uma programação flexível dos semáforos ao longo da semana.

4. **Exibir Status dos Semáforos:** O Controlador Semafórico possui a funcionalidade de visualizar o status atual dos semáforos em tempo real. Isso permite ao controlador monitorar o funcionamento dos semáforos, identificar possíveis falhas ou problemas de operação e tomar medidas corretivas adequadas.

Esses casos de uso representam as principais funcionalidades do sistema de controle semafórico, proporcionando ao Controlador Semafórico a interação com as informações relacionadas aos intervalos de tempo, perfis de intervalo luminoso e intervalos de dias, além de visualizar o status em tempo real dos semáforos.

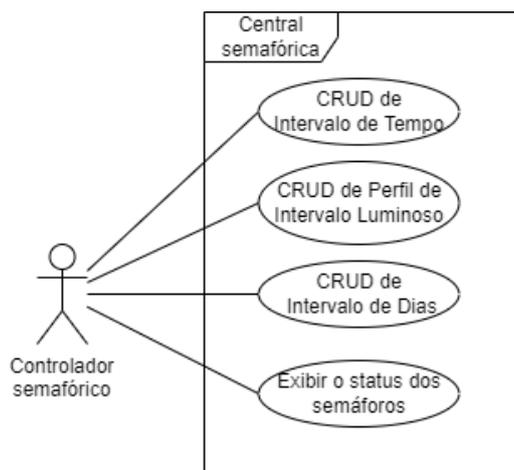


Figura 4. Modelo de caso de uso
Fonte: Elaborado pelo próprio autor - 2023

7. Diagrama de entidade relacionamento

De acordo com [Franck et al. 2021], um diagrama de entidade-relacionamento (ER) "é um modelo popular para a modelagem conceitual de dados. Esse modelo e suas variações, como o Modelo Entidade-Relacionamento Estendido, são frequentemente usados para o projeto conceitual de aplicações de banco de dados." O DER representado na Figura 5 do sistema de Controle Semafórico é dividido em quatro entidades distintas.

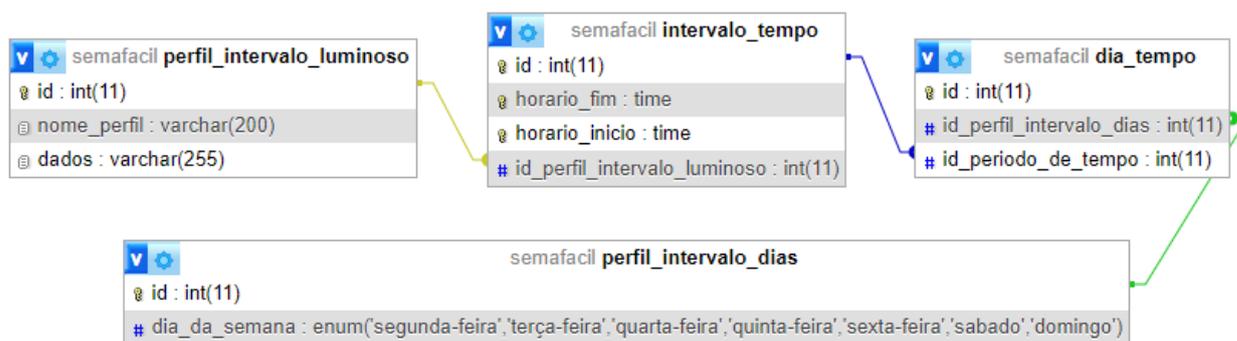


Figura 5. Diagrama de entidade relacionamento
Fonte: Elaborado pelo próprio autor - 2023

A primeira entidade, denominada "Perfil de Intervalo Luminoso", possui um total de 3 atributos.

A entidade "Perfil Intervalo Luminoso" tem como principal objetivo armazenar as informações de execução das luzes semafóricas e seus tempos, com os seguintes atributos:

- "id" (chave primária): Identifica exclusivamente cada perfil de intervalo luminoso.
- "nome-perfil": Responsável por armazenar o nome do perfil do intervalo luminoso.
- "dados": Contém informações específicas do perfil de intervalo luminoso.

Essa informação específica refere-se ao comportamento em si de ambos os semáforos A e B e é estruturada da seguinte forma:

- O início da estrutura contém uma identificação única para o semáforo, como "SEMA" (Semáforo A) ou "SEMB" (Semáforo B).
- Cada fase de operação do semáforo é representada por um conjunto de letras e números. As letras representam as cores, R: Vermelho, G: Verde, Y: Amarelo e iY: Amarelo intermitente.
- Os números representam a duração em segundos da fase correspondente. A ordem das letras indica a sequência de exibição das cores.
- Um caractere especial é "-" pode ser usado para separar as configurações de diferentes fases ou semáforos.

Exemplo genérico:

-SEMA R 10 G 15 Y 5 -SEMB G 20 R 8

Neste exemplo genérico:

1. **SEMA** é o identificador do primeiro semáforo.
2. R 10: Vermelho por 10 segundos.
3. G 15: Verde por 15 segundos.
4. Y 5: Amarelo por 5 segundos.
5. - indica a transição para o próximo semáforo.
6. **SEMB** é o identificador do segundo semáforo.
7. G 20: Verde por 20 segundos.
8. R 8: Vermelho por 8 segundos.

A segunda entidade, designada como "Intervalo de Tempo", armazena informações relacionadas ao horário de execução de um perfil específico de intervalo luminoso. Seus atributos são: "id"(chave primária), que identifica cada intervalo de tempo; "horario-inicio", sendo o início da operação do semáforo no intervalo de tempo correspondente; "horario-fim", registrando o término da operação do semáforo e "id-perfil-intervalo-luminoso"(chave estrangeira), que estabelece uma conexão com o "Perfil Intervalo Luminoso"ao qual o intervalo de tempo está associado.

Quanto à terceira entidade, intitulada "Perfil Intervalo Dias", ela registra os dias nos quais as operações semafóricas serão aplicadas. Seus atributos são: "id"(chave primária), que identifica exclusivamente cada perfil de intervalo de dias; "dia-da-semana", representando qualquer dia da semana (de Segunda a Domingo); e "id-intervalo-tempo"(chave estrangeira), estabelecendo uma referência ao "Intervalo de Tempo"ao qual o intervalo de dias pertence.

A quarta entidade, atuando como uma ponte entre "Perfil Intervalo Dias"e "Intervalo de Tempo", permite a relação N:M entre as duas entidades. Denominada "Dia Tempo", essa tabela inclui os atributos: "id"(chave primária), "id-perfil-intervalo-dias"(chave estrangeira referenciando "Perfil Intervalo Dias") e "id-periodo-de-tempo"(chave estrangeira referenciando "Intervalo de Tempo").

8. Materiais

Na seção "Materiais"serão detalhadas as ferramentas, linguagens de programação, banco de dados e *hardware* utilizados no protótipo do sistema Semafácil. Além disso, serão apresentados os custos associados ao projeto. As informações serão abordadas em subtópicos para proporcionar uma visão detalhada.

8.1. Hardware

Nesta seção, serão apresentados os componentes de *hardware* para a implementação do sistema Semafácil, juntamente com uma tabela de custos que detalha todos os equipamentos utilizados. O projeto Semafácil é composto pelos seguintes itens:

- *Raspberry Pi Zero*: Um *single-board computer* utilizado como o núcleo do sistema.
- Fonte de Alimentação: Uma fonte de 5 volts e 2.2A para fornecer energia ao *Raspberry Pi Zero*.
- Cartão de Memória: Um cartão de memória de 32GB utilizado para armazenar o sistema operacional e outros dados essenciais.

- Dois Semáforos em Miniatura: Componentes visuais que representam os semáforos no ambiente de teste do projeto.

Esses elementos desempenham papéis cruciais na operação do Semafácil, e a tabela de custos fornecerá uma visão detalhada dos investimentos associados a cada componente.

8.1.1. Raspberry Pi Zero

O *Raspberry Pi* é um computador de placa única de baixo custo desenvolvido pela *Raspberry Pi Foundation*, com o propósito de oferecer um dispositivo acessível para projetos educacionais. O *Raspberry Pi* destaca-se por seu tamanho compacto e grande potencial de ensino, bem como pela capacidade de solucionar problemas no âmbito da *IoT*. Conforme mencionado no blog [Open Source Community 2023], "Pessoas em todo o mundo utilizam o Raspberry Pi para adquirir habilidades de programação, desenvolver projetos de hardware, automatizar residências, implementar clusters *Kubernetes* e computação de borda, e até mesmo empregá-los em contextos industriais." A *Raspberry Pi Foundation* oferece diversos modelos de computadores de placa única, sendo o modelo escolhido para o desenvolvimento do projeto o *Raspberry Pi Zero W*.

O *Raspberry Pi Zero W* é um dos modelos de baixo custo e excelente custo-benefício da marca, contendo as seguintes configurações essenciais para o projeto [Foundation 2023]:

- Suporte nativo ao sistema *open-source Linux Raspbian* - Permitindo ampla utilização de soluções feitas para o Linux.
- Processador de 1GHz, single-core com 512MB de RAM - Ideal para tarefas com pouca exigência de processamento.
- Conexão *Wireless LAN 802.11 b/g/n* - Permitindo o acesso remoto e configuração do sistema.
- *Header GPIO* com 40 pinos - Interface ideal para fazer a comunicação entre *Raspberry Pi* e as luzes semafóricas.

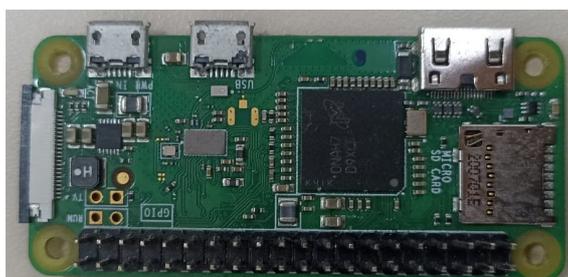


Figura 6. Raspberry Pi Zero W
Fonte: Elaborado pelo próprio autor - 2023

8.1.2. Luzes semafóricas

Os semáforos utilizados no sistema Semafácil consistem em dois conjuntos de LEDs, sendo cada conjunto composto por uma LED vermelha, uma amarela e uma verde. Cada semáforo possui 4 pinos, um para cada cor e outro para o *GND* (*Ground*), conforme representado na imagem 7 abaixo.



Figura 7. Semáforo em miniatura - Semafácil
Fonte: Elaborado pelo próprio autor - 2023

8.1.3. Custos do projeto

Para a confecção do projeto, foram necessários os seguintes itens: um *single-board computer*, escolhido como o *Raspberry Pi Zero WH*. Esse modelo foi selecionado devido à sua alta versatilidade para projetos *IoT*. Para o seu funcionamento, foram adquiridos um cartão de memória de 32GB e uma fonte compatível de 5V 2.2 amperes.

Para a representação semafórica, foram comprados dois módulos semafóricos, cada um com três LEDs (um vermelho, um amarelo e um verde, representados na figura 7), além dos fios necessários para conectar os módulos semafóricos ao *Raspberry Pi*.

Os preços, em reais, apresentados na tabela 3, foram pesquisados em 01/12/2023. No entanto, é possível encontrar ofertas melhores em *e-commerces* e plataformas chinesas.

É importante salientar que o presente projeto é um protótipo. Soluções finais, com *hardware* diferenciado, terão um custo maior.

Quantidade	Hardware	Preço sem o valor do frete
1x	Raspberry Pi Zero WH	R\$ 251,71
1x	Cartão de memória 32GB	R\$ 29,90
1x	Fonte 5v 2.2A	R\$ 39,90
2x	Módulo semafórico	R\$ 12,50
1x	Cabos Jumper Fêmea X Fêmea 40 pcs	R\$ 15,29
TOTAL		R\$ 361,80

Tabela 3. Tabela de custos - Preços pesquisados no dia 01/12/2023

8.2. Software

Durante o projeto, foram consideradas diversas ferramentas e linguagens para o desenvolvimento, tanto para o *back-end* quanto para o *front-end*. No *front-end* do sistema, uma interface *web* foi desenvolvida utilizando *HTML* para a estruturação das páginas [Mozilla 2023b], *CSS* para a estilização [Mozilla 2023a], e *JavaScript* para a adição de elementos complexos com atualização dinâmica [Mozilla 2023c]. Além disso, foram empregadas as bibliotecas *Bootstrap*, que fornecem um conjunto de ferramentas e estilos CSS para o desenvolvimento de páginas responsivas [W3schools 2023], e *jQuery*, uma biblioteca rica em recursos para o *JavaScript*, funcionando em diversos navegadores e ampliando a gama de ferramentas disponíveis para programação *web* [JQUERY 2023]. Outro recurso de suma importância para o projeto foi o *stepped-bar-chart*, um gráfico de barras escalonadas desenvolvido em *JavaScript* pelo programador independente *nyash04* [NYASH04 2023]. O uso simultâneo de arquivos *JSON* e banco de dados no projeto atende a propósitos distintos. Os arquivos *JSON* são exclusivamente utilizados para alimentar a biblioteca *stepped-bar-chart*, a qual requer esse formato para exibir gráficos de barras escalonadas no *front-end*. Dessa forma, é necessário traduzir constantemente os dados armazenados no banco de dados para o formato *JSON*, possibilitando a representação visual gráfica dos diagramas de intervalo luminoso.

Já o *back-end* do sistema é composto pelo *PHP*, tendo seu código executado pelo servidor e gerando um *HTML* que é enviado para o usuário [PHP 2023], e o *Python*, linguagem orientada a objetos com tipagem dinâmica excelente para o desenvolvimento rápido de aplicações [Python 2023]. O *PHP* foi utilizado para diversas tarefas, como cadastrar, remover e editar informações do banco de dados, manipular arquivos do tipo *JSON*, responsáveis pela troca leve de dados, sendo de fácil utilização tanto pelas máquinas quanto pelos humanos [JSON 2023], gerar botões, controlar os serviços do *software* Semafácil, entre outras tarefas. Já o *Python* foi primariamente utilizado para fazer a conexão entre as luzes semafóricas e o *Raspberry Pi Zero WH*, além de monitorar o status das luzes semafóricas em tempo real através do *framework Flask* (módulo *Python* que permite desenvolver facilmente aplicações *web* [Flask 2023]), consultando informações do banco de dados e enviando comandos para as luzes em tempo real, utilizando a biblioteca *RPi.GPIO*, que "fornece um módulo *Python* para controlar o *GPIO*" [Pypi 2023].

8.2.1. Banco de Dados

O banco de dados selecionado para o sistema foi o *MySQL*. Ele adota um modelo de banco de dados relacional, um tipo que armazena e possibilita o acesso a conjuntos de dados inter-relacionados [ORACLE 2023]. O *MySQL* armazena dados em tabelas, aplicando regras para os relacionamentos, garantindo assim a integridade e consistência dos dados [MYSQL 2023]. O banco de dados foi projetado com base no Diagrama de Entidade-Relacionamento (DER), representado na figura 5.

⁶*GPIO* significa Entrada/Saída de Propósito Geral. Esses pinos são uma interface física entre o *Raspberry Pi* e o mundo externo, [Raspberry 2023]

```
import RPi.GPIO as GPIO
import time
import argparse
import signal
import sys

# Função para ligar um LED específico
def ligar_led(pino, duracao):
    GPIO.output(pino, True)
    time.sleep(duracao)
    GPIO.output(pino, False)
```

```
semafacil@raspberrypi:/var/www/html/old/old/Semafacil $
python3 programa.py -SEMA 5 4 6 -SEMB 4 2 3
```

Figura 8. Imports e a função ligar led do primeiro programa ao lado dos parâmetros de execução

9. Desenvolvimento

Inicialmente, além da pesquisa bibliográfica, realizou-se o levantamento de requisitos do sistema (Tabela 2) e a elaboração de um diagrama arquitetural (Figura 3), os quais foram fundamentais para as etapas subsequentes.

O projeto desenvolvido foi desenvolvido em duas etapas. A primeira etapa, parcialmente representada na figura 8, foi direcionada para a criação da primeira versão do *software*, capazes de controlar as luzes semafóricas, e configuração e instalação das dependências do projeto. Já a segunda etapa foi direcionada para a implementação e refino dos requisitos funcionais representados na Tabela 2.

9.1. Primeira etapa

A primeira parte do projeto destacou-se pela montagem do *Raspberry Pi Zero*, instalação e configuração do *Raspbian*, sistema operacional do *Raspberry Pi*, incluindo dependências do projeto, como o *Apache Web Server*, um servidor *HTTP* moderno para sistemas operacionais *Windows* e *Linux* [Apache 2023], e *Python*, entre outras. Com a conclusão da preparação do ambiente e dos sistemas, deu-se início à etapa de integração entre as luzes semafóricas e o *Raspberry Pi*.

Durante a integração, foi desenvolvido um *script* em *Python* que utiliza a biblioteca *RPi.GPIO* para realizar a comunicação com os pinos *GPIO* do *Raspberry Pi*, acendendo e apagando os LEDs em tempo real. O programa recebia os dados passados em parâmetros pelo terminal do *Linux*, por exemplo: **python3 programa.py -SEMA 5 4 6 -SEMB 4 2 3**, onde **-SEMA** e **-SEMB** definiam qual semáforo deveria ser executado, e os números 5, 4, 6 próximos ao **-SEMA** definiam o tempo em segundos de execução das cores "verde, amarela e vermelha", nessa ordem específica. Também era possível determinar se uma cor iria piscar de forma intermitente adicionando um "i" na frente dos segundos, Ex: **python3 programa.py -SEMA 4i 4 6 -SEMB 4i 2 3**, nesse exemplo, ambas as cores verdes dos semáforos piscariam de forma intermitente por 4 segundos.

Com a conclusão da integração, o próximo objetivo tornou-se a apresentação do sistema de forma gráfica, permitindo que o usuário enviasse comandos aos semáforos sem a necessidade de acessar o sistema operacional diretamente. Para isso, foi desenvolvida uma página *web* utilizando o *Apache* e *Bootstrap*. O principal objetivo dessa página era receber informações e enviá-las por meio de um *back-end PHP* para o programa *Python*, que, por sua vez, controlaria o comportamento dos semáforos.



Figura 9. Primeira interface do programa ao lado de seu *back-end* PHP - 2023

Após a conclusão da primeira etapa, deu-se início ao processo de desenvolvimento do sistema Semafácil em si, utilizando os diagramas previamente elaborados. Contudo, identificou-se um problema na lógica do sistema que comprometeria a expansão da primeira etapa, tornando inutilizados tanto o *back-end* em *Python* quanto o *back-end* em *PHP*.

O programa inicialmente desenvolvido em *Python* não atendia aos requisitos do sistema. Ambos os semáforos executavam seus comandos sequencialmente, em vez de simultaneamente, o que inviabilizava a criação de uma programação semafórica realista. Além disso, não era possível definir uma ordem de execução personalizada para os semáforos, limitando-os a seguir uma sequência fixa de verde, amarelo e vermelho. Esses problemas específicos seriam abordados na próxima etapa do projeto, visando expandir o sistema e alinhá-lo com os princípios pré-definidos.

9.2. Segunda etapa

Na segunda etapa do projeto, deu-se início à correção dos problemas identificados na primeira etapa, subdividindo-os nos seguintes tópicos: Desenvolvimento de um programa em *Python* capaz de executar informações simultâneas para ambos os semáforos do sistema. Adicionalmente, efetuou-se a implementação de diversos *CRUDs*, como Perfil de Intervalo Luminoso, Intervalo de Tempo e Intervalo de Dias. Além disso, criou-se um programa para a integração dos semáforos com as informações do banco de dados, possibilitando a exibição em tempo real dos semáforos conforme descrito na tabela 2. Houve também a integração de todas as funcionalidades em uma única interface *web* e a adição de controle para o início e parada dos métodos.

9.2.1. Execução simultânea dos semáforos

Para realizar a execução simultânea das luzes semafóricas através de parâmetros, foi adotado o uso de *threads*⁷ no programa *Python*. Além do uso de *threads*, permitiu-se que

⁷Threading em *Python* permite que diferentes partes do seu programa sejam executadas simultaneamente e pode simplificar o design do seu código. [RealPython 2023]



Figura 10. Tela Perfil de Intervalo Luminoso - 2023

o usuário defina a ordem de execução das luzes semafóricas relacionadas, através de um único parâmetro, como mencionado no Tópico 7, proporcionando um maior controle sobre o comportamento das luzes semafóricas.

9.2.2. Perfil de Intervalo Luminoso

Com a implementação da execução simultânea dos semáforos concluída, deu-se início ao desenvolvimento do *CRUD* de Perfil de Intervalo Luminoso. Esse processo envolve a criação, leitura, atualização e exclusão das informações no banco de dados responsáveis por programar o funcionamento das luzes semafóricas. Como resultado, a tela final apresentada na Figura 10 é composta pelos seguintes elementos: a exibição de dois semáforos e suas configurações atuais, utilizando e adaptando a biblioteca *stepped-bar-chart* de [NYASH04 2023], uma entrada para o nome do perfil de intervalo luminoso, um botão para adicionar informações no banco de dados, um "Painel de edição dos semáforos" e uma tabela exibindo o ID, nome do perfil de intervalo luminoso e as ações que podem ser realizadas nas informações, tais como um botão "Excluir" atualmente presente.

No "Painel de Edição de Semáforos", Figura 11, é apresentada a página onde é possível editar a programação dos semáforos antecipadamente, antes de enviá-las para a base de dados. Nessa tela, há a prévia de dois semáforos para serem editados, o "Semáforo A" e o "Semáforo B". Cada prévia mostra o tempo de execução das cores, permitindo uma inspeção ao passar o mouse sobre elas.

Na modificação individual de cada semáforo, conforme exibido na Figura 13, apresenta-se a programação do semáforo. A listagem individual de cada cor exibe o nome, a duração em segundos e se é intermitente ou não, com opções para editar (ver Fi-

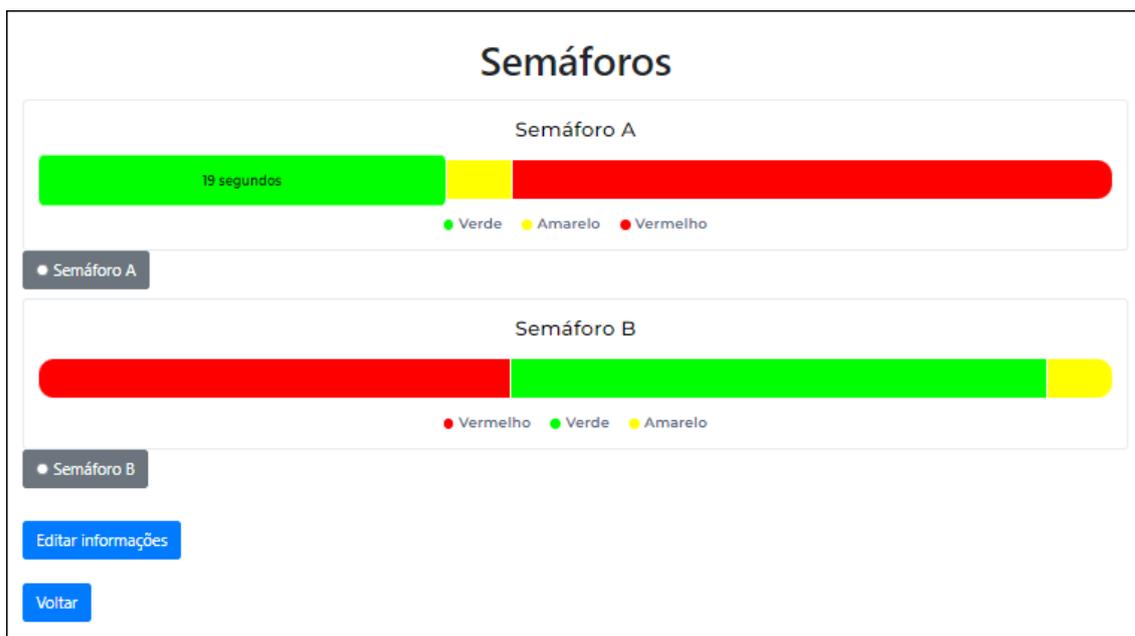


Figura 11. Painel de edição de semáforos - 2023

Figura 12. Tela para edição da cor - 2023

gura 12) ou excluir a cor. Adicionalmente, há um submenu para adicionar uma nova cor, com opções para verde, vermelho e amarelo, e a escolha de tornar a cor intermitente. O submenu inclui um botão "Adicionar" e outro para retornar à página anterior, denominado "Voltar".

Nesse momento, ambos os semáforos manipulados estão sendo consultados a partir de dois arquivos *JSON*, **SEMA.json** e **SEMB.json**, salvos no *Raspberry Pi*. Esses arquivos *JSON* são utilizados pela biblioteca *stepped-bar-chart* para a exibição gráfica dos semáforos. Portanto, além de salvar, editar, exibir e remover as informações dos arquivos *JSON*, é necessário convertê-las para os parâmetros do tipo **-SEMA R G Y -SEMB R G Y** para o banco de dados e vice-versa.

O arquivo *JSON* contém um ID único, o nome da cor, o valor em segundos, o código da cor e se ela possui o comportamento intermitente.

Modificação do semáforo - A

Semáforo A



● Verde ● Amarelo ● Vermelho

Lista de cores

Nome da cor	Segundos	Cor	Intermitente	Ação
Verde	19			Editar Excluir
Amarelo	3			Editar Excluir
Vermelho	28			Editar Excluir

Adicionar nova cor

Cor:

Segundos:

Intermitente:

Figura 13. Edição dos semáforos individualmente - 2023

9.2.3. Intervalo de tempo

Após a conclusão do desenvolvimento do "Perfil de Intervalo Luminoso", deu-se início ao desenvolvimento do *CRUD* de Intervalo de Tempo. O objetivo desse *CRUD*, representado na Figura 14, é cadastrar, ler, atualizar e editar as informações relacionadas aos horários nos quais o "Perfil de Intervalo Luminoso" será executado. Portanto, o formulário inclui campos para "Horário de Início", "Horário de Fim" e uma opção para selecionar o "Perfil de Intervalo Luminoso". Além disso, há uma tabela exibindo as informações atualmente cadastradas, com botões para editar e excluir cada entrada.

ID	Horário de Início	Horário de Fim	ID do Perfil	Ações
7	12:00	23:59	Diagrama de intervalo luminoso	Editar Excluir

Figura 14. Tela de Intervalo tempo - 2023

Alguns tratamentos de exceção também foram adicionados ao cadastro de "Perfil de Intervalo Luminoso", não permitindo que o usuário cadastre um horário de início maior que o horário de fim, exibindo uma mensagem de erro em forma de *popup*.

9.2.4. Intervalo de Dias

Com a finalização dos dois principais *CRUDs*, "Perfil de Intervalo Luminoso" e "Intervalo de Tempo", deu-se início ao último *CRUD* necessário para o sistema: o "Gerenciamento do Intervalo de Dias". Seu objetivo é determinar em quais dias os semáforos irão funcionar. A tela, representada na Figura 15 é constituída pelos seguintes elementos: uma tabela com o ID do cadastro do Intervalo de Dias com: dia da semana, o horário de início e fim da tabela de intervalo de tempo, o nome do perfil de intervalo luminoso que executa no dia e dois botões, um para editar e o outro para remover, além do botão "Adicionar Novo Dia Tempo" que redireciona para uma tela de cadastro.

A tela de cadastro do "Intervalo de Dias", demonstrada na imagem 16, permite que o usuário selecione o dia desejado e o período de tempo, com o intervalo luminoso, que será aplicado naquele dia. A tela possui um botão para "Adicionar Dia Tempo" e outro para retornar à página anterior.

O cadastro das informações no *back-end* exigiu cuidado especial, uma vez que a inclusão de uma informação deveria evitar que o usuário inserisse dados de execução

Dia Tempo					
ID	Dia da Semana	Horário Início	Horário Fim	Nome do Perfil	Ações
26	sabado	12:00:00	23:59:00	Diagrama de intervalo luminoso	<input type="button" value="Editar"/> <input type="button" value="Remover"/>

Figura 15. Tela de Gerenciamento de Dias - 2023

Adicionar Novo Dia Tempo

Perfil de Intervalo de Dias:

segunda-feira

Período de Tempo:

12:00:00 - 23:59:00

Figura 16. Tela de cadastro de Intervalo de Dias - 2023

conflitantes na tabela. Para lidar com essa questão, implementou-se um *trigger*⁸ no banco de dados. O *trigger* associado à tabela "dia-tempo", verifica a existência de entradas no banco de dados com períodos de tempo sobrepostos para o mesmo perfil de intervalo de dias. Se um conflito é identificado, o *trigger* gera um sinal de erro, impedindo a operação e garantindo a integridade dos dados relacionados aos semáforos.

9.3. Programa de Integração dos Semáforos com as Informações do Banco de Dados

Com a conclusão de todos os *CRUDs* do sistema, surgiu a necessidade de criar um programa capaz de ler as informações do banco de dados e aplicá-las apenas durante seus respectivos períodos de tempo. Assim, foi desenvolvido o serviço denominado **integração-banco-semaforo.py**. O programa em *Python* interage com o banco de dados para verificar, em intervalos regulares, se o horário atual, antecipadamente consultado através do sistema operacional do Raspberry Pi que utiliza um servidor NTP⁹ de endereço *a.ntp.br* (*Network Time Protocol*), corresponde a algum perfil de intervalo luminoso. Quando ocorre uma correspondência, ele executa uma lógica associada aos semáforos. A verificação acontece em um loop infinito, e a execução é feita por meio do sistema desenvolvido no Tópico 9.2.1, utilizando subprocessos para lidar com a lógica do semáforo e *GPIO* para interação com os pinos no *Raspberry Pi*. O programa é projetado para ser executado em segundo plano, porém também pode ser acionado via terminal, verificando continuamente e ajustando o estado dos semáforos conforme necessário.

⁸Procedimento armazenado no banco de dados que é automaticamente acionado sempre que ocorre um evento específico no banco de dados. [Alura 2023]

⁹*NTP* significa *Network Time Protocol* ou Protocolo de Tempo para Redes. É o padrão que permite a sincronização dos relógios dos dispositivos de uma rede como servidores, estações de trabalho, roteadores e outros equipamentos a partir de referências de tempo confiáveis.[NTP.BR 2023]

```
semafacil@raspberrypi: /var/www/html
semafacil@raspberrypi: /var/www/html $ python3 integracao_banco_semaforo.py
Conexão bem-sucedida!
Conexão bem-sucedida!
{'id': 28, 'dia_da_semana': 'terça-feira', 'horario_inicio': datetime.time(19, 3, 28), 'horario_fim': datetime.time(19, 3, 28), 'nome_perfil': 'Diagrama de intervalo luminoso', 'dados_perfil_luminoso': '-SEMA G 19 Y 3 R 28 -SEMB R 22 G 25 Y 3 '}
Nenhuma correspondência encontrada para o horário atual.
```

Figura 17. Execução do integração-banco-semaforo.py via terminal - 2023

Se acionado via terminal, é possível observar o programa realizando seu processo de execução, como ilustrado na Figura 17. Durante uma varredura no sistema, o programa carrega todas as configurações semafóricas, aguardando para executá-las no momento em que os horários correspondentes são alcançados. Enquanto não há script para ser executado, o sistema exibe a mensagem "Nenhuma correspondência encontrada para o horário atual" a cada segundo.



Figura 18. Status das luzes semafóricas - 2023

9.4. Exibir Status dos Semáforos em Tempo Real

Para exibir o status dos semáforos em tempo real, utilizou-se o *framework Flask*, mencionado anteriormente, que atua como um servidor *web*, monitorando o comportamento

dos semáforos a cada segundo e enviando informações em tempo real para o sistema Semafácil. A página "Status dos semáforos" apresenta seis informações distintas, cada uma representando uma luz semafórica de um semáforo específico, conforme ilustrado na imagem 18. Quando desativadas, os elementos são exibidos em vermelho, enquanto, quando ativados, os elementos ficam verdes. Na Figura 18, todas as luzes estão desativadas.

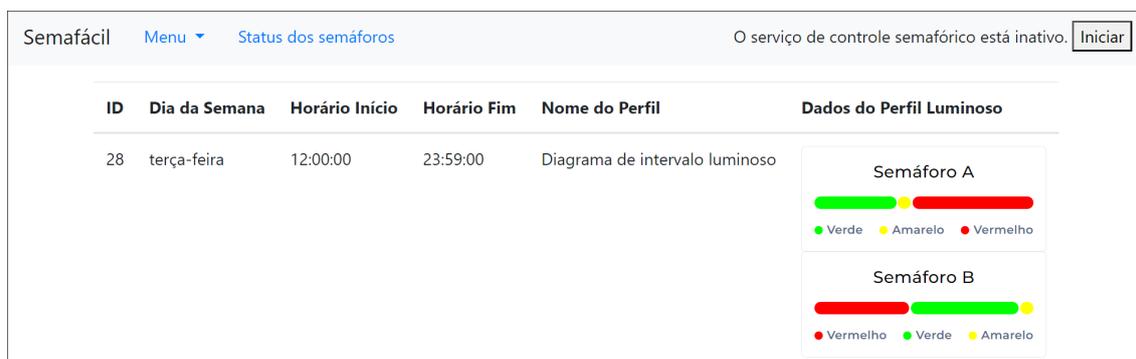


Figura 19. Interface Web Integrada - 2023

9.5. Unificação das Funcionalidades na Interface Web Integrada

Com os principais métodos do projeto concluídos, o próximo passo foi consolidar todas as funcionalidades em uma única interface, acessível e de fácil compreensão para o usuário. A interface do sistema Semafácil, observada na Figura 19, é composta por: uma *nav-bar*¹⁰, presente em todas as demais páginas, contendo um menu que possibilita a transição entre os *CRUDs* de manipulação de intervalo luminoso, intervalo de tempo e intervalo de dia. Adicionalmente, há um botão para acessar a página de visualização do status dos semáforos, assim como um botão para "Iniciar" ou "Desligar" o serviço semafórico, acompanhado de seu respectivo status. Abaixo, encontra-se uma tabela com o ID, dia da semana, horário de início, horário de término, nome do perfil de intervalo luminoso e os dados do perfil de intervalo luminoso apresentados em um gráfico de barras escalonadas (*stepped-bar-chart*), proporcionando uma visão geral dos *scripts* em funcionamento no sistema.

10. Conclusão

O desenvolvimento do projeto Semafácil representou um desafio significativo e uma enorme fonte de estímulo para o aprendizado e reflexão, empregando grande parte dos conhecimentos adquiridos ao longo do curso de Tecnologia em Análise e Desenvolvimento de Sistemas. O projeto partiu de um ponto inicial totalmente desprovido de estrutura para se tornar um sistema concreto e tangível.

Em perspectiva a futuras atualizações, manifesta-se a possibilidade em ampliar a capacidade do sistema para gerenciar múltiplos semáforos. Planeja-se também a implementação de um controle de acesso para usuários. Outro ponto relevante para evolução é a introdução de câmeras e sensores no sistema, visando aprimorar o controle de tráfego e pedestres. Essa incorporação tecnológica proporcionará uma visão mais

¹⁰Link para seções/páginas apropriadas em um site que auxilia os leitores na navegação pelo documento online. [TECHNOPEDIA 2023]

abrangente e precisa do ambiente, permitindo ajustes dinâmicos e eficazes nas condições de tráfego.

Referências

- Alura (2023). O que é e como usar trigger em sql. [Online; Acessado em 02 de dezembro de 2023].
- Apache (2023). Apache servidor http. [Online; Acessado em 02 de dezembro de 2023].
- Ashton, K. (1999). That internet of things. *RFID journal*, 22.
- BARROS (2019). Introdução à teoria fuzzy.
- CONTRAN (2014). Volume v: Sinalização semafórica. https://www.gov.br/infraestrutura/pt-br/assuntos/transito/arquivos-senatran/educacao/publicacoes/manual_vol_v_2.pdf. [Online; Acessado em 5 de abril de 2023].
- Dandil, E. and Gültekin, S. (2017). Web service-based automation system for duration scheduling and remote control of traffic signal lights. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 948–953.
- dos Transportes, M. (2023). Sobre o contran. [Online; Acessado em 18 de dezembro de 2023].
- Flask (2023). O que é o flask python. [Online; Acessado em 02 de dezembro de 2023].
- Foundation, R. P. (2023). What is a raspberry pi? <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. [Online; Acessado em 23 de maio de 2023].
- Franck, K. M., Pereira, R. F., and Filho, J. V. D. (2021). Diagrama entidade-relacionamento: uma ferramenta para modelagem de dados conceituais em engenharia de software. *Research, Society and Development*, 10:e49510817776.
- Gokhale, P., Bhat, O., and Bhat, S. (2018). Introduction to iot. *International Advanced Research Journal in Science, Engineering and Technology*, 5(1):41–44.
- IEEE (2015). Towards a definition of the internet of things (iot). https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf. [Online; Acessado em 20 de junho de 2023].
- Isikdag, U. (2015). *Internet of Things: Single-Board Computers*, pages 43–53. Springer International Publishing, Cham.
- JQUERY (2023). What is jquery? [Online; Acessado em 01 de dezembro de 2023].
- JSON (2023). Introdução ao json. [Online; Acessado em 01 de dezembro de 2023].
- Kanagachidambaresan, G. (2021). *Role of Single Board Computers (SBCs) in rapid IoT Prototyping*. Internet of Things. Springer International Publishing.
- MOZILLA (2023). O que é crud? <https://developer.mozilla.org/pt-BR/docs/Glossary/CRUD>. [Online; Acessado em 22 de maio de 2023].
- Mozilla (2023a). O que é css? [Online; Acessado em 01 de dezembro de 2023].

Mozilla (2023b). O que é html? [Online; Acessado em 01 de dezembro de 2023].

Mozilla (2023c). O que é javascript? [Online; Acessado em 01 de dezembro de 2023].

MYSQL (2023). O que é o mysql? [Online; Acessado em 01 de dezembro de 2023].

NTP.BR (2023). O ntp. [Online; Acessado em 21 de dezembro de 2023].

NYASH04 (2023). Stepped bar chart. [Online; Acessado em 01 de dezembro de 2023].

Onn, A. B., Salim, S., Ahmad, M. S., and Jamil, M. M. A. (2014). Development of a network-enabled traffic light system. *Proceedings - 4th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2014*, pages 241–244.

Open Source Community (2023). What is a raspberry pi? Acessado em 25 de maio de 2023.

ORACLE (2023). O que é rdbms (banco de dados relacional)? [Online; Acessado em 01 de dezembro de 2023].

Otero, M. T. L. (2022). Control experimental de semáforos vía web con raspberry pi y pitraffic.

PHP (2023). O que é o php. [Online; Acessado em 01 de dezembro de 2023].

Pypi (2023). Módulo de controle python para raspberry pi. [Online; Acessado em 01 de dezembro de 2023].

Python (2023). O que é o python. [Online; Acessado em 01 de dezembro de 2023].

Python¹ (2023). Interface python para tcl/tk. [Online; Acessado em 02 de dezembro de 2023].

Raspberry (2023). Pinos gpio. [Online; Acessado em 02 de dezembro de 2023].

RealPython (2023). Uma introdução a threads em python. [Online; Acessado em 02 de dezembro de 2023].

SOMMERVILLE, I. S. (2016). *Engenharia de software*. Pearson Higher Education, vol.10 edition.

TECHNOPEDIA (2023). O que é uma navigation bar? [Online; Acessado em 03 de dezembro de 2023].

W3schools (2023). O que é bootstrap. [Online; Acessado em 01 de dezembro de 2023].

Documento Digitalizado Público

Artigo Final - Pedro J. Gilo

Assunto: Artigo Final - Pedro J. Gilo
Assinado por: Rodolfo Oliveira
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Rodolfo Francisco de Oliveira, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 21/12/2023 19:12:00.

Este documento foi armazenado no SUAP em 21/12/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1533797

Código de Autenticação: 3e8a341876

