



Universidade Federal de Santa Catarina  
Centro Tecnológico  
Departamento de Engenharia Elétrica

**Projeto Final:**

**Atualização da documentação, código fonte aplicativo  
para smartphone e demonstração do funcionamento**

**Disciplina:**

EEL7323 – Programação C++ para Sistemas Embarcados

**Alunos:**

Leonardo Baltazar Clivati (19100814)

Guilherme Henrique Paggi Daros (19100811)

**Professores:**

Eduardo Augusto Bezerra

Anderson Wedderhoff Spengler

**Turma:**

08235

**Florianópolis, 19 de dezembro de 2022.**

# 1. Introdução

O projeto segue a ideia da proposta apresentada, consiste de uma estação para o monitoramento da temperatura do ambiente. A estação é constituída por um microcontrolador e um sensor de temperatura, além disso, contém um modelo que prevê o próximo valor e registra caso ocorra um acerto. Em complemento, ao sistema microcontrolado é apresentado um programa para computador e um aplicativo, para observação e armazenamento dos dados, sendo a comunicação com o primeiro realizada por meio da serial e a segunda com uso de um protocolo de comunicação sem fio.

O objetivo é fazer a parte inicial de um projeto que permita a economia de energia na transmissão de informação do sistema meteorológico, pelo fato de que se o dado predito estiver próximo do real, o envio não seria necessário. Além disso, esse escopo permite a aplicação de diferentes conceitos apresentados ao longo da disciplina e envolve o desenvolvimento e comunicação entre diferentes plataformas.

## 2. Ferramentas de desenvolvimento

Primeiramente, para o desenvolvimento da estação de monitoramento é utilizada uma placa NUCLEO-F401RE, que contém um microcontrolador da ST, e apresentada compatibilidade com aplicações de *machine learning*. Ainda, para o monitoramento da temperatura é acoplado a placa MPU-6050, que contém sensores de temperatura, aceleração e giroscópio, esse foi escolhido apenas pela disponibilidade, um módulo que monitorasse outros parâmetros, como temperatura e umidade, seria mais adequado para essa aplicação. A comunicação entre o sensor e o microcontrolador é feita por meio do protocolo I2C e a comunicação com o computador é feita utilizando a UART.

Para o desenvolvimento do código do módulo é empregada a plataforma STM32CubeIDE, a qual permite uma configuração simples dos pinos do controlador e oferece as bibliotecas base para a comunicação I2C e UART, além de possuir uma interação mais simples com a placa escolhida. A comunicação UART pode ser feita utilizando a porta USB, pela qual o microcontrolador é programado, isso deve facilitar o desenvolvimento dessa etapa. No quesito da comunicação sem fio, o professor recomendou em aula o uso do Bluetooth, assim, será planejado a integração do módulo Bluetooth BLE V4.0 HM-10 Keyes, o qual permite interação por meio de UART.

Vale mencionar que para o software do computador, o desenvolvimento é feito Visual Studio Code e compilação é estruturada utilizando CMake, sendo quase todo o código elaborado do zero, com algumas funções utilitárias do Linux. Por fim, o programa para o celular é elaborado utilizando o Android Studio, o projeto será feito com base no template da aplicação “Hello World” oferecida na plataforma, pelo fato do foco da aplicação estar na parte do *back-end*, o *front-end* será uma tela simples.

### 3. Fluxograma

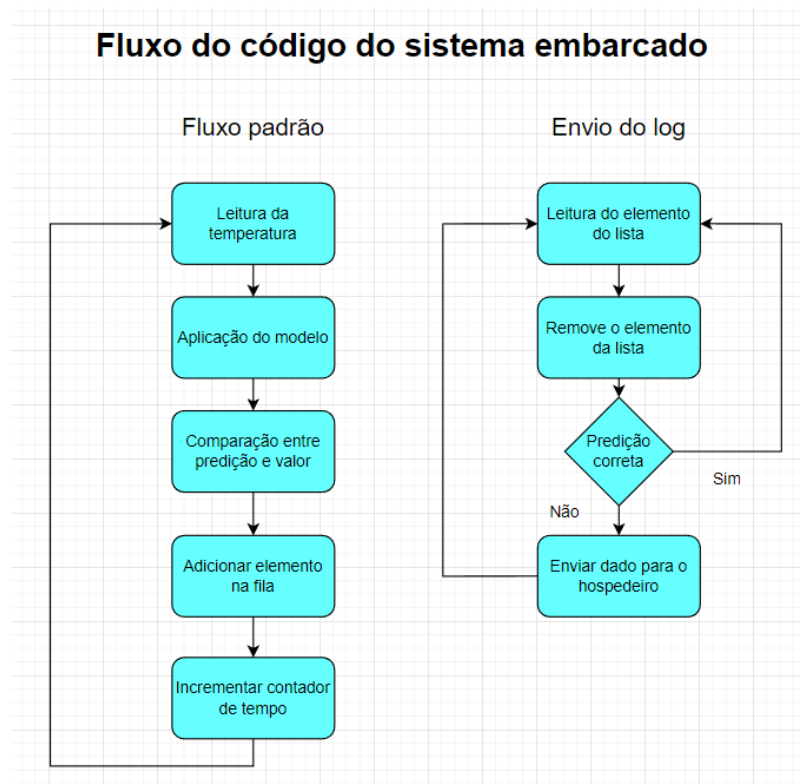


Figure 1 - Fluxograma do sistema embarcado

### 4. Máquina de estados

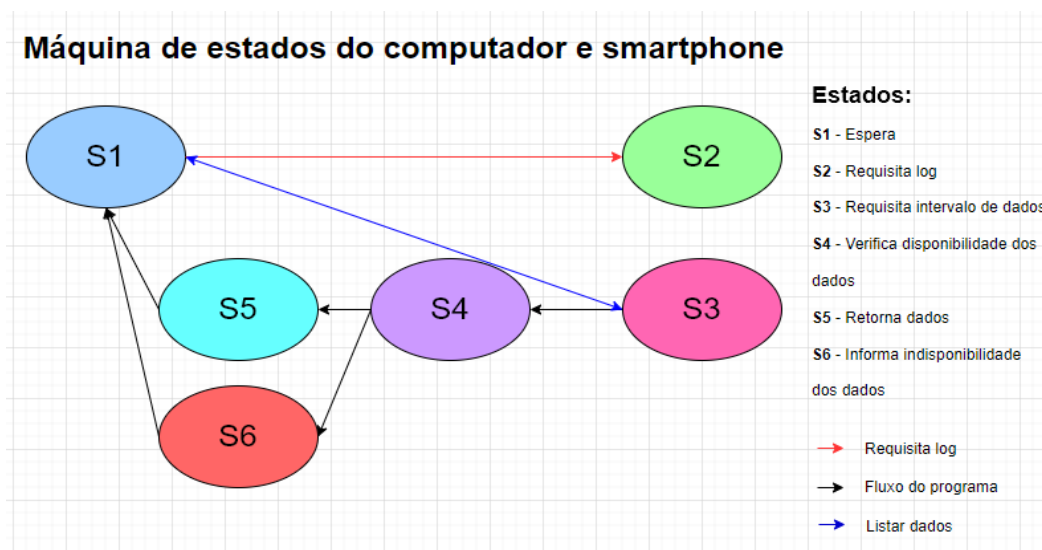


Figure 2 - Máquina de estado do programa para computador e smartphone

## 5. Diagrama de classes

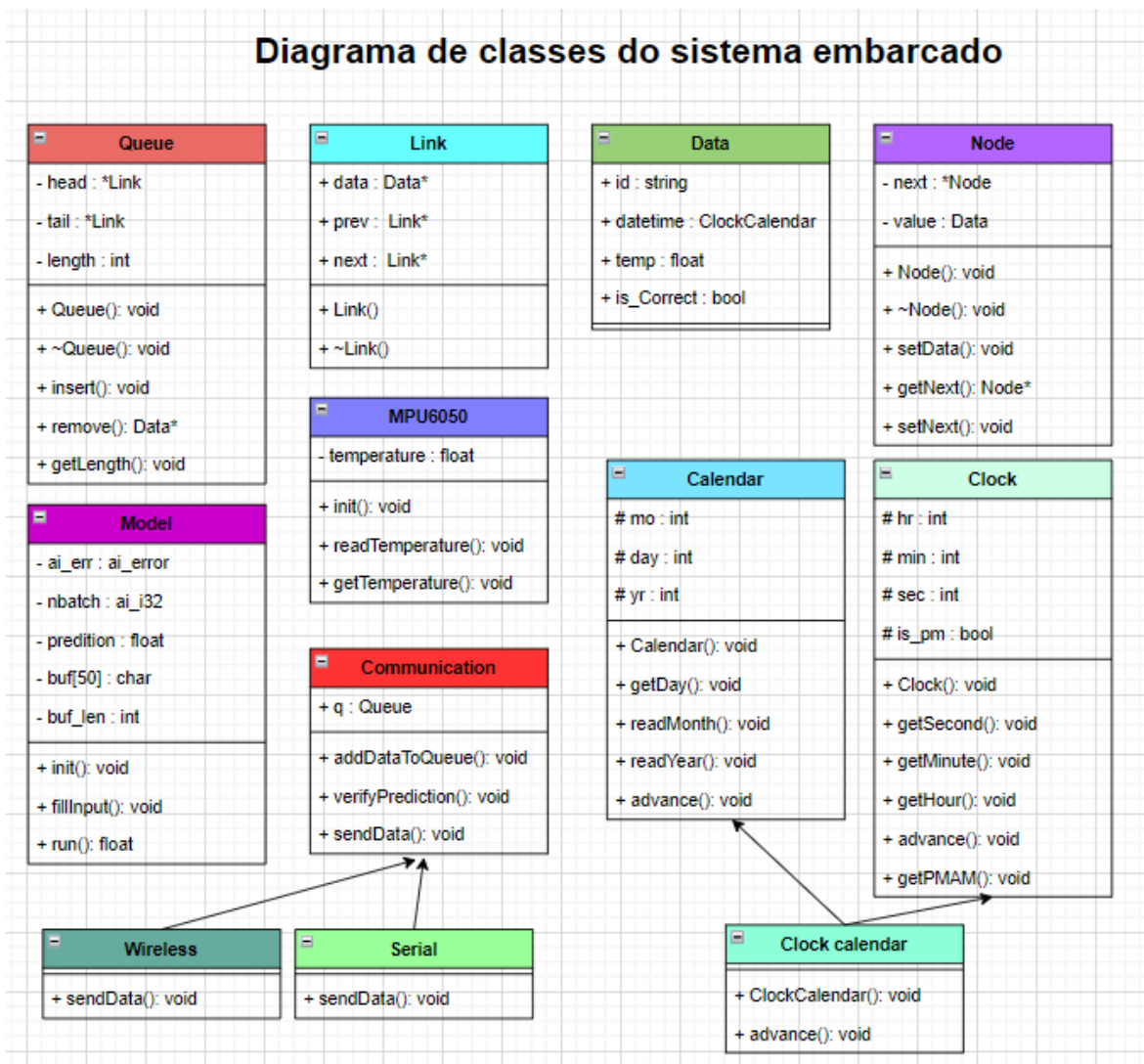


Figure 3 - Diagrama de classes do sistema embarcado

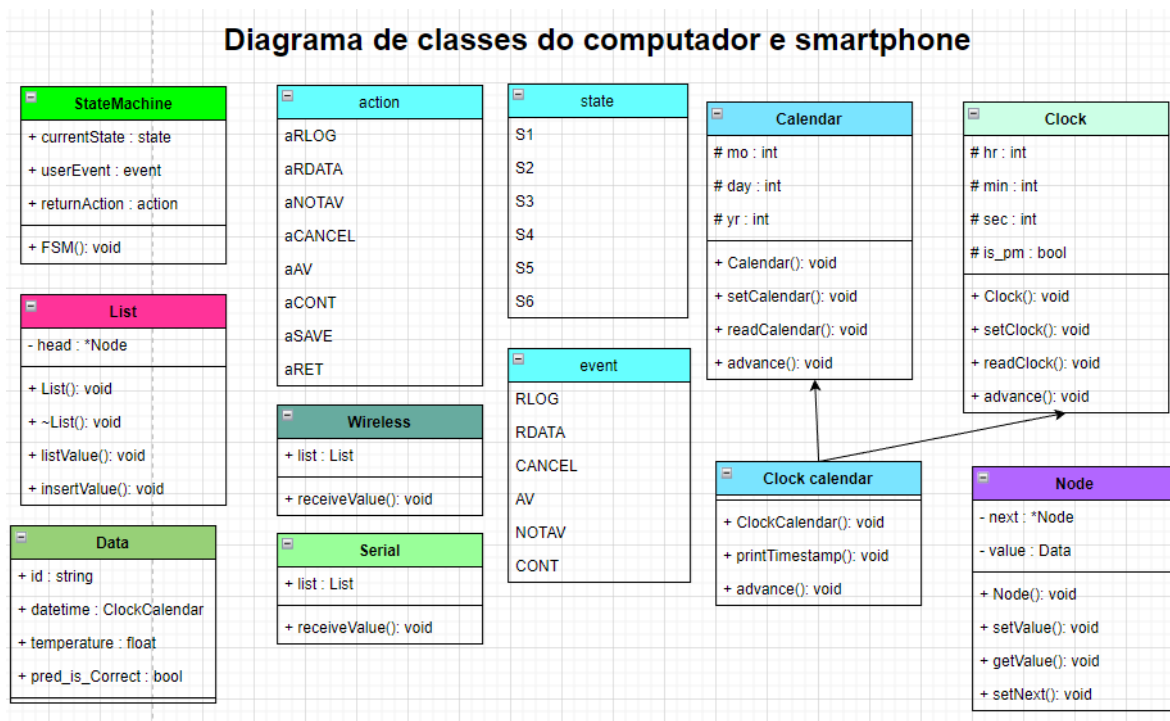


Figure 4 - Diagrama de classes do programa do computador e smartphone

## 6. Modelagem do sistema

O programa embarcado segue o fluxo de esperar as medidas do sensor, realizar a predição, seguir para comparação e após isso armazenar os resultados em uma fila. As informações salvas são a temperatura medida, se o modelo acertou a predição, o código de identificação do módulo e a *timestamp*. Ainda, é implementada a funcionalidade de transmitir os dados pela serial ou sem fio, utilizando o conceito de polimorfismo, com a subsequente reinicialização da fila. Todo esse processo pode ser observado no fluxograma e no diagrama de classes.

Em relação à parte de envio do log, foi criada uma superclasse Communication para concentrar as funções e parâmetros comuns a comunicação serial e wireless, e uma função virtual pura que está relacionada a transferência dos dados. A estrutura empregada para o armazenamento é a fila, pelo fato de permitir um envio ordenado dos dados do mais antigo para o mais recente, de modo a tornar o armazenamento ordenado no tempo no hospedeiro. Enquanto, no hospedeiro a estrutura de dados utilizada foi a lista, na qual é possível acessar diferentes nodos sem as restrições presentes na fila e na pilha.

Ainda, vale ressaltar que, no código da fila foram implementadas as funções para adicionar e remover elementos, além de requisitar o tamanho da estrutura. Com o intuito de otimizar o programa, o método para remover nodos da fila também retorna seus valores, de maneira que é possível enviar a mensagem e apagá-la em apenas um passo. O tamanho da fila é empregado no laço de repetição que envia os dados com uso da UART.

O programa do computador inicializa e espera uma conexão serial, esta foi abstraída por uma classe, com a estação, assim que detectada o programa orienta o usuário a apertar um botão que sinaliza ao software embarcado que o envio de dados deve ser realizado. O computador recebe os dados em formato json, faz o *parsing* e insere na lista ligada os eventos em ordem. Em seguida o programa pede ao usuário que insira no terminal dois *timestamps*, e em seguida realiza uma busca na lista ligada, retornando todos os dados entre os dois *timestamps* que o usuário inseriu. Por fim o programa do computador pergunta ao usuário se ele deseja fazer uma nova busca ou terminar o programa.

No aplicativo, a interface será composta por dois botões, um para solicitar os dados da estação e outro para listar o log. Ademais, na função de listar o log é necessário colocar dois campos para adicionar o período de dados que quer ser visualizado. Em relação ao modelo de ML, os dados serão extraídos do sensor e o treinamento realizado no Google Colab, e posteriormente a implementação no microcontrolador, a ideia é fornecer nove medidas para o modelo e tentar verificar se o décimo valor e a predição estão próximo, com margem de até 0,5°C. Para a comunicação Bluetooth, o método de comunicação será implementado com base no datasheet do módulo, porém o teste não será possível pelo fato do estudante não possuir o componente.

A implementação do modelo é realizada com uso da biblioteca proprietária da ST, *Cube IA*, a qual facilita a integração de modelos de *machine learning* em microcontroladores da empresa. A configuração é realizada por meio da interface gráfica da IDE e possui a vantagem de utilizar menos memória que a aplicação com uso do *tensorflow*, de acordo com material fornecido pela *Digikey*. O modelo foi implementado na classe *model* e realiza a predição conforme o planejado.

## 7. Plano de teste

Primeiramente, os sistemas devem ser testados separadamente, assim, no microcontrolador será demonstrado a funcionalidade de envio dos dados e realização da predição pelo modelo, com auxílio da ferramenta de *debug* presente na ST32CubeIDE. O aplicativo será testado somente de maneira isolada, dessa forma, será elaborada uma função que gera amostras aleatórias na estrutura de dados utilizada no projeto, ao ser pressionado o botão que requisita os dados para a estação meteorológica. Ainda, será testada a funcionalidade que mostra os dados de acordo com o período inserido, datas fora do padrão serão informadas para confirmar a robustez do sistema.

O programa de computador, primeiramente, será alimentado com dados provenientes de um microcontrolador que já possui a fila de eventos em memória na inicialização, apenas esperando pela requisição dos dados para realizar o envio. Novamente, nesse caso serão inseridas datas que estão fora do padrão a fim de mostrar a robustez do código. Na sequência poderá ser realizada a interação entre o código do computador e o sistema embarcado, sendo realizada a requisição dos dados e posterior transferência. Por fim, para verificar a capacidade de armazenamento dos códigos será gerado um loop para adicionar o máximo de elementos nas estruturas de dados, fila e lista, e em seguida serão criadas funções para impedir que isso ocorra.

## 8. Resultados

Para atestar o funcionamento do software do computador foi realizado um teste com um microcontrolador que já é inicializado com uma fila de eventos e apenas fica esperando o botão ser apertado para enviar os dados via serial. Com isso pode-se validar a recepção dos dados no computador, o *parsing* utilizando formato *json* e a inserção na fila de eventos do software do computador. O algoritmo de busca no tempo também foi validado, para isso utilizou-se de uma função semelhante a *getTimeFromUserMenu*, porém sem a necessidade de input do usuário em *runtime*, *getTimeFromCodeMenu*, para testar diversos casos, todos validados, atestando o seu funcionamento.

Em relação ao microcontrolador, o fluxo padrão de funcionamento do programa foi testado, ou seja, a aquisição dos dados por meio do sensor, a realização da predição da temperatura, comparação entre os valores e finalmente o registro na fila. Ainda, o envio dos dados também foi atestado com auxílio de um visualizador da comunicação serial, e os resultados de ambos os testes ocorreram de acordo com o esperado. O programa utiliza 6,11 kB (6,36%) de memória RAM e 39,9kB (7,79%) da flash da placa de desenvolvimento, assim, a implementação pode ser considerada leve.

Os vídeos comprovando o funcionamento dos dois códigos podem ser encontrados no [drive](#) e código fonte da aplicação comentado no [repositório do github](#).

## 9. Comentários

Devido ao acúmulo de trabalho no final do semestre e a falta de conhecimento dos integrantes da equipe na elaboração de aplicativo, não foi possível finalizar a aplicação para o celular. Ainda, para a integração satisfatória entre o código do microcontrolador e do computador seria necessário modificar a estrutura da fila do sistema embarcado para ficar semelhante à do outro software e permitir comunicação por meio da serial.

Os conceitos requisitados na especificação do projeto foram todos implementados com sucesso e podem ser encontrados nos dois programas, exceto o tratamento de exceções. Por fim, levando em consideração a grande carga de atividades propostas no trabalho, a entrega final da equipe ficou em um nível satisfatório e possibilitou a aplicação de conceitos ensinados ao longo do semestre e novos aprendizados, como a aplicação de *machine learning* em sistemas embarcados.