



# **Sistema de monitorização contínua, não-invasiva, para pacientes internados em enfermaria**

## *Relatório de Projeto I*

**Aluno:** Joao Pedro Afonso Coutinho

**Matrícula:** 1421768

**Orientadora:** Noemi Rodriguez

**Período:** 2020.1

Rio de Janeiro

Dezembro/2020

## Sumário

<b>1. Introdução</b> .....	3
<b>2. Contexto Atual</b> .....	4
<b>Propostas e Objetivos do Trabalho</b> .....	5
<b>3. Especificação do Sistema</b> .....	6
<b>3.1 Módulo Sensor</b> .....	6
<b>3.2 Estação Base</b> .....	7
<b>4. Desenvolvimento</b> .....	8
<b>4.1 Microcontroladores</b> .....	8
<b>4.2 Sensores</b> .....	10
<b>5. Atividades realizadas</b> .....	12
<b>5.1 Plano de ação original</b> .....	12
<b>5.1.1 Cronograma original</b> .....	14
<b>5.2 Revisão do plano de ação</b> .....	14
<b>5.2.1 Atividades realizadas</b> .....	15
<b>5.3 Discussão entre o que foi proposto e o que foi feito</b> .....	15
<b>6. Conclusão</b> .....	17
<b>7. Referências bibliográficas</b> .....	18

# 1. Introdução

A avaliação de sinais vitais como frequência cardíaca (FC), frequência respiratória (FR), saturação parcial de oxigênio (SpO2) e temperatura são essenciais para determinar a gravidade e a progressão de diversas doenças.

Atualmente nas enfermarias de hospitais, o monitoramento dos pacientes depende da rotina dos enfermeiros e auxiliares. Esse processo manual limita a frequência de monitoramento, o que pode não atender o tempo de resposta para os casos de mudanças rápidas do estado clínico, como o caso da COVID-19.

Nesse projeto propomos um sistema de monitoramento automático de baixo custo, não-invasivo, que avalia o paciente periodicamente através de um módulo sensor e envia os dados e possíveis alertas para uma estação base remota.

O projeto está sendo desenvolvido em um ambiente Linux, e envolve a programação dos microcontroladores de 32 bits *nRF52832*, da *Nordic Semiconductor* e *ESP8266*, da *Espressif Systems*. O firmware do *nRF52832* é desenvolvido em C++, utilizando o *Software Development Kit (SDK) 17*, fornecido pela própria fabricante. Para compilar e gravar o firmware no chip utilizamos a IDE *Segger Embedded Studio (SES)*.

O firmware para o *ESP8266* está sendo desenvolvido em Lua, sobre a plataforma open source *NodeMCU*. Utilizamos a IDE *ESPlorer* para gravar o firmware no microcontrolador.

## 2. Contexto Atual

Atualmente, apenas os pacientes internados em Unidades de Tratamento Intensivo (UTI), devido a gravidade de seu estado de saúde, recebem monitoramento contínuo de seus indicadores de saúde. Esse monitoramento se dá através dos caros monitores multiparamétricos (Figura 1), equipamento médico que chega a custar milhares de reais.



Figura 1 - Monitor multiparamétrico

Diferentemente da UTI, pacientes internados em enfermaria não estão em estado crítico de saúde e, devido ao elevado custo dos monitores multiparamétricos, são monitorados apenas em intervalos regulares, que variam entre duas e oito horas. Esse monitoramento requer que o profissional da saúde vá até o paciente e afira alguns de seus sinais vitais manualmente.

No contexto da pandemia, doenças como a COVID-19, na qual pacientes comumente não percebem estarem com um baixo nível de oxigênio no sangue (hipoxemia), o simples monitoramento contínuo da SpO2 poderia identificar o agravamento das condições clínicas desses pacientes, diminuindo o tempo de resposta dos profissionais da saúde. Nestas situações o monitoramento intervalado se mostra deficiente pois não consegue prever súbitas alterações de estado clínico.

Casos como este nos fazem perceber a importância do monitoramento contínuo do paciente internado em enfermaria, a fim de reduzir riscos e antecipar ações emergenciais.

## **Propostas e Objetivos do Trabalho**

Propomos um sistema utilizando hardware facilmente encontrado no mercado e software a ser desenvolvido como parte desse projeto, a fim de reduzir os custos e possibilitar o uso massivo do projeto em enfermarias.

A proposta desse trabalho não é substituir a utilização dos monitores multiparamétricos em hospitais, mas sim oferecer uma alternativa de monitoração a pacientes internados em enfermaria, que já não são monitorados continuamente.

O objetivo deste trabalho é desenvolver um sistema de baixo custo e não-invasivo que possibilite o monitoramento contínuo dos sinais vitais de pacientes internados em enfermaria e a transmissão destes dados em intervalos de tempo programáveis. Esse monitoramento visa à rápida detecção às mudanças de estado clínico destes pacientes, possibilitando uma igualmente rápida tomada de decisão por parte dos profissionais da saúde responsáveis.

### 3. Especificação do Sistema

O sistema é composto por duas partes: um módulo sensor, responsável por aferir a temperatura, a frequência cardíaca e a saturação parcial de oxigênio do paciente, e uma estação base remota, responsável por receber os dados clínicos dos pacientes, transmitidos pelos módulos sensores, identificar esses dados e disponibilizá-los para serem acessados pela internet. Essa interação ocorre conforme o diagrama da Figura 2:

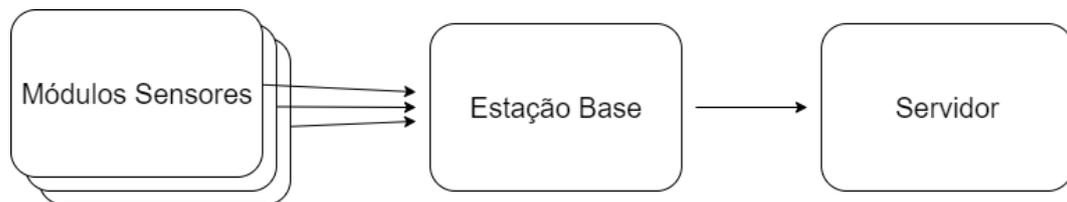


Figura 2- Diagrama do sistema

#### 3.1 Módulo Sensor

O módulo sensor é composto por um microcontrolador *nRF52832*, da *Nordic Semiconductor* e dois sensores, um de temperatura e outro de SpO<sub>2</sub>, com frequência cardíaca integrado. O sensor de temperatura é o *MLX90614*, fabricado pela *Melexis*, já o sensor de SpO<sub>2</sub> é o *MAX30102*, da fabricante *Maxim Integrated*.

Os sensores se comunicam com o microcontrolador através do protocolo I<sup>2</sup>C. Seguindo o protocolo, o *nRF5282* é configurado como Master e os sensores *MLX90614* e *MAX30102* como Slave (Figura 3).

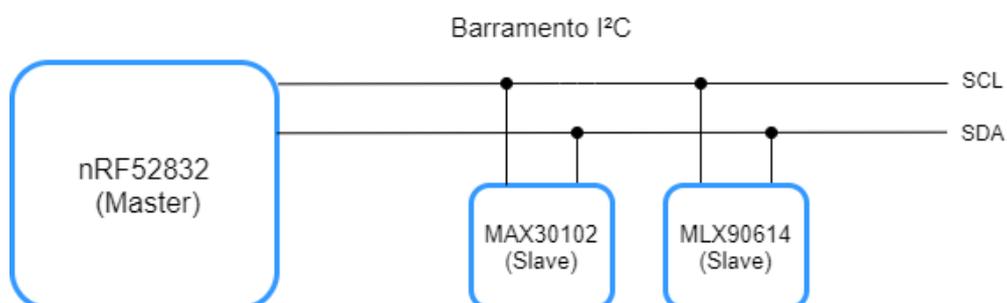


Figura 3 – Comunicação entre sensores e microcontrolador

Uma característica interessante do nRF52832 é que o mesmo já possui um rádio de 2.4GHz embutido, rádio esse utilizado para transmitir os dados obtidos à estação base através do protocolo proprietário da Nordic, o *Enhanced ShockBurst*.

### 3.2 Estação Base

A estação base é composta por dois microcontroladores, um responsável pelas funcionalidades de internet, o ESP8266, e outro responsável pela recepção dos dados transmitidos via rádio de um módulo sensor, o nRF52832. A comunicação entre os chips se dá através do protocolo SPI, conforme a Figura 4.



Figura 4 - Comunicação entre o rádio e o chip Wi-Fi

Para a comunicação entre a estação base e o servidor utilizamos o protocolo de troca de mensagens MQTT.

## 4. Desenvolvimento

O projeto está sendo desenvolvido com o foco em dois pontos: baixo custo e disponibilidade de componentes. Todos os componentes utilizados no projeto podem ser facilmente encontrados no mercado e estão na faixa dos 3 dólares.

Nas subseções abaixo apresentamos melhor esses componentes.

### 4.1 Microcontroladores

#### 4.1.1 nRF52832

*System on Chip* (SoC) de propósito geral da *Nordic Semiconductor* que integra uma CPU Arm® Cortex™-M4, com unidade de ponto flutuante, e um transceptor RF de 2,4GHz. Possui memória flash de 512 KB e RAM de 64 KB.

O nRF52832 (Figura 5) é um SoC multiprotocolo que suporta Bluetooth 5, ANT e diversos protocolos proprietários, como o *Enhanced ShockBurst* (ESB), que é utilizado neste projeto.

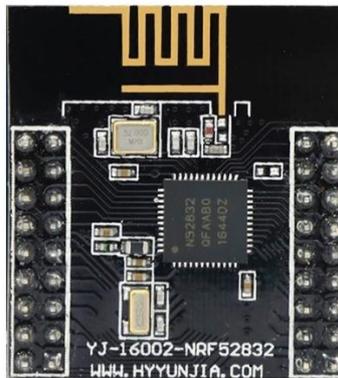


Figura 5 - nRF52832 em placa de desenvolvimento

Além do transceptor RF, o chip possui inúmeros periféricos, tais como conversores analógico-digital, timers, NFC, e interfaces como SPI, UART e I<sup>2</sup>C. O diagrama de blocos simplificado do SoC é apresentado na Figura 6.

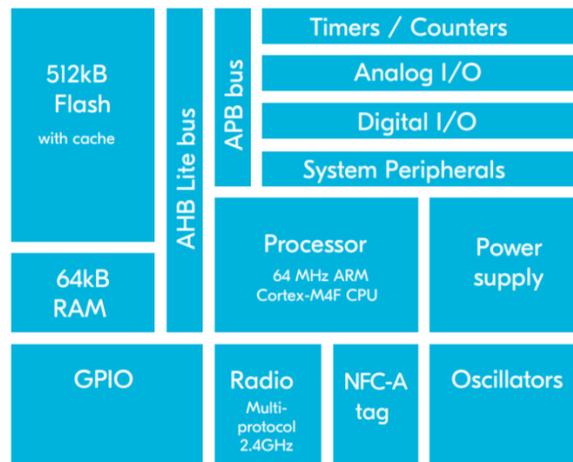


Figura 6 - Diagrama de blocos simplificado do nRF52832

O nRF52832 opera numa frequência de 64 MHz e conta com um excepcional baixo consumo de energia, alcançado através de um sofisticado sistema interno que gerencia o consumo automaticamente. Aliado a isto o microcontrolador opera com tensões de 1.7 V a 3.6V, o que possibilita seu funcionamento com baterias e pilhas sem maiores complicações.

#### 4.1.2 ESP8266:

*System on Chip* (SoC) da fabricante *Espressif System* que integra uma CPU RISC de 32-bits Tensilica L106, e um transceptor RF de 2,4GHz, utilizado para conexões à redes Wi-Fi. Possui memória flash externa de 512 KB, expansível até 16 MB, e RAM de 64 KB.



Figura 7 - ESP8266 na placa de desenvolvimento Weemos D1 Mini

A principal característica do ESP8266 é suportar os padrões de rede sem fio 802.11 b/g/n. No padrão 802.11n (2.4 GHz) o chip suporta taxas de transmissão de até 72.2 Mbps.

Assim como o nRF52832, o chip também possui inúmeros periféricos, tais como conversores analógico-digital, timers, e interfaces como SPI, UART e I<sup>2</sup>C. O diagrama de blocos simplificado do SoC é apresentado na Figura 8

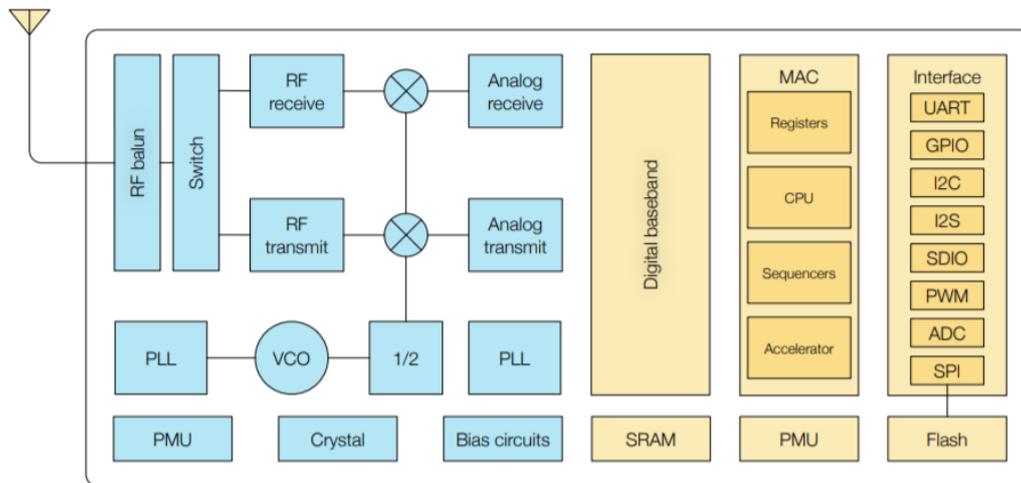


Figura 8 - Diagrama de blocos do ESP8266

O ESP8266 opera na frequência de 80 MHz por padrão, podendo operar também em 160MHz, e trabalha com tensões de 2.5 V a 3.6V, com uma corrente de operação média de 80mA.

## 4.2 Sensores

### 4.2.1 Sensor de SpO2 e frequência cardíaca

Neste projeto utilizamos o *MAX30102*, da fabricante *Maxim Integrated* (Figura 9), para oximetria de pulso e monitoramento de frequência cardíaca.



Figura 9 - MAX30102

Esse sensor apresenta uma interface de comunicação compatível com o padrão I<sup>2</sup>C, opera com tensões de 3.1V a 5.0V e exige baixa corrente, na faixa dos 600 $\mu$ A a 1200 $\mu$ A.

#### 4.2.2 Sensor de temperatura

O *MLX90614*, da Melexis (Figura 10), é um sensor infravermelho de alta precisão, não sendo necessário contato com o objeto mensurado.



Figura 10 – Sensor de temperatura infravermelho MLX90614

Vem calibrado de fábrica para medir temperaturas entre -70°C e +380°C, porém sua precisão de  $\pm 0,5^\circ\text{C}$  só é alcançada para objetos com temperatura entre 0°C e +50°C.

Assim como o MAX30102, esse sensor apresenta uma interface de comunicação compatível com o padrão I<sup>2</sup>C, e possui versões de 3.0V e 5.0V.

## 5. Atividades realizadas

As atividades realizadas durante o Projeto Final I seguiram o plano de ação apresentado na Proposta de Projeto Final. Esse mesmo plano pode ser lido abaixo, na subseção *Plano de ação original*.

Nas subseções seguintes temos a *Revisão do plano original*, que inclui a descrição das atividades realizadas durante o período, e temos uma *Discussão entre o que foi proposto e o que foi feito*.

### 5.1 Plano de ação original

Primeiramente precisamos identificar as necessidades do projeto. Essa parte do plano de ação foi parcialmente concluída.

Precisamos testar os componentes pré-selecionados para o módulo sensor e garantir que eles atendem às necessidades do projeto. Precisamos garantir que o *System on Chip (SoC) nRF52832* atenderá de forma adequada às necessidades do módulo sensor. Precisamos nos assegurar também que as leituras dos sensores *MAX30102* e *MLX90614* são confiáveis. Para isso podemos comparar as leituras desses sensores com a leitura de equipamentos médicos garantidamente aferidos.

Precisaremos também compreender melhor o Software Development Kit (SDK) do *nRF52832* e desenvolver as bibliotecas de comunicação entre os sensores e o SoC.

Como o projeto envolve coleta e análise de sinais vitais, precisamos pesquisar na literatura médica a forma como esses sinais vitais são interpretados para a definição do estado clínico de um paciente. Precisamos pesquisar também os ajustes das leituras dos sensores a depender da parte do corpo do paciente onde os sinais vitais foram aferidos.

Em um segundo momento precisamos testar os componentes pré-selecionados para a estação base e garantir que eles atendem às necessidades do projeto. Precisamos garantir que o SoC *ESP8266* atenderá de forma adequada às necessidades da estação base. Precisamos nos assegurar que o *ESP8266* conseguirá se comunicar com o *nRF52832* e que conseguirá se conectar e

transmitir as informações recebidas pela rede Wi-Fi. Estudar o desenvolvimento de código para o ESP8266 utilizando a plataforma NodeMCU.

Após isso, precisamos iniciar a integração do sistema. Precisamos definir o protocolo de comunicação do módulo sensor com a estação base e o protocolo de comunicação entre a estação base e a internet. Nesta etapa é necessário estudar também sobre a segurança dos dados transmitidos. Como estamos trabalhando com informações médicas, o sistema deverá possuir um mínimo de segurança quanto aos dados transmitidos ou pelo menos possibilitar a implementação de alguma forma de segurança.

Com o sistema já integrado podemos iniciar a fase de testes do sistema. Os testes deverão verificar o desempenho do sistema quanto a capacidade de transmissão dos dados e consumo de energia.

O produto final criado será um sistema composto de pelo menos duas partes: o módulo sensor e a estação base. Ambas as partes do sistema podem ser subdivididas em eletrônica e software, desenvolvidas paralelamente. O produto final também contém toda a documentação do sistema e os resultados dos testes.

Sendo possível dentro do prazo do projeto, uma terceira etapa poderá ser adicionada. Estando o protótipo funcional e devidamente testado, poderemos focar na parte estética do projeto, como desenvolver uma caixa para abrigar o protótipo e uma interface gráfica para exibir os dados transmitidos pela estação base. Também poderemos, por se tratar de um sistema médico, realizar testes de segurança. Esta terceira etapa está condicionada ao funcionamento do protótipo ao término de Projeto Final I.

Proponho que este plano de ação seja seguido através de ciclos de desenvolvimento. Cada um dos parágrafos acima seria desenvolvido em um ciclo de aproximadamente duas semanas, totalizando quatro meses para o desenvolvimento do projeto. Cada ciclo prevê ao menos uma reunião com a professora orientadora do projeto.

Para Projeto Final I teríamos então a documentação e um protótipo funcional para os testes de integração e as discussões de segurança. Para Projeto Final II ficaríamos os testes de funcionamento do protótipo, finalização da documentação iniciada em Projeto Final I. Qualquer alteração que o protótipo, não percebida durante Projeto Final I será feita em Projeto Final II.

### 5.1.1 Cronograma original

Para o plano de ação original foi proposto o cronograma apresentado na Figura 11.

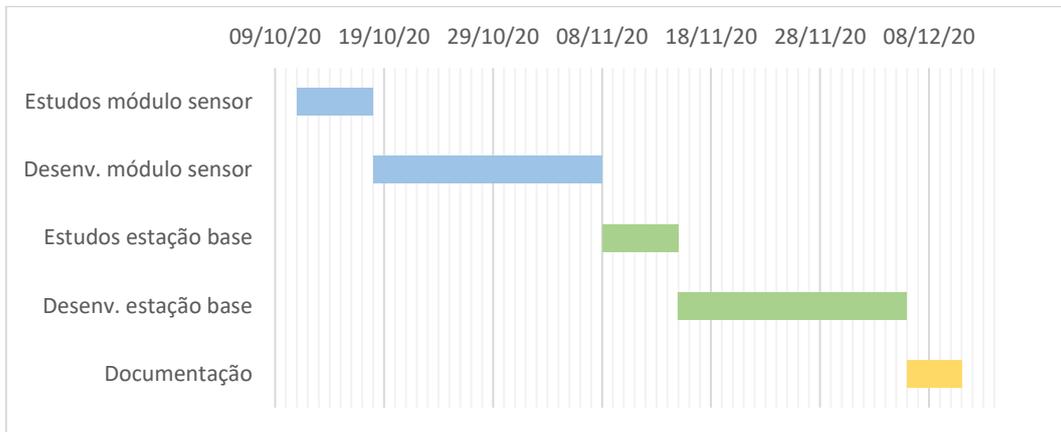


Figura 11 - Cronograma original

### 5.2 Revisão do plano de ação

O plano de ação não sofreu mudanças drásticas. Como tarefa de desenvolvimento do modulo sensor atrasou por motivos a serem discutidos na subseção 5.3, as tarefas posteriores foram adiadas conforme o cronograma da Figura 12:

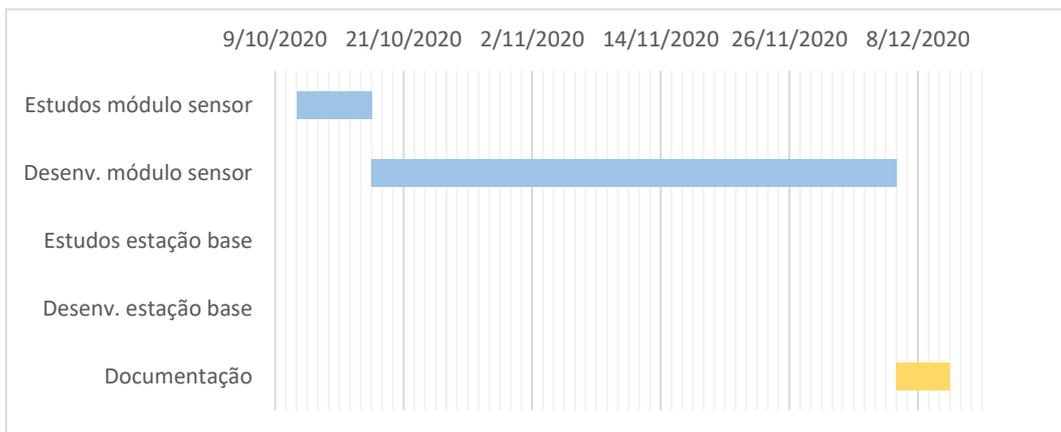


Figura 12 - Cronograma atualizado

### 5.2.1 Atividades realizadas

As atividades realizadas foram todas dedicadas ao desenvolvimento do módulo sensor.

Estudei o SDK 17 da Nordic Semiconductor para conseguir desenvolver um firmware para SoC nRF52832.

Todos os sensores foram testados e suas bibliotecas foram adaptadas para funcionarem com o nRF52832. Além disso desenvolvi o código para obtenção e processamento dos dados aferidos pelos sensores.

Parte da comunicação com a estação também foi desenvolvida. O módulo sensor consegue enviar via rádio proprietário do SoC os dados obtidos pelos sensores.

Um primeiro protótipo do módulo sensor foi montado. O protótipo pode ser visto na Figura 13:

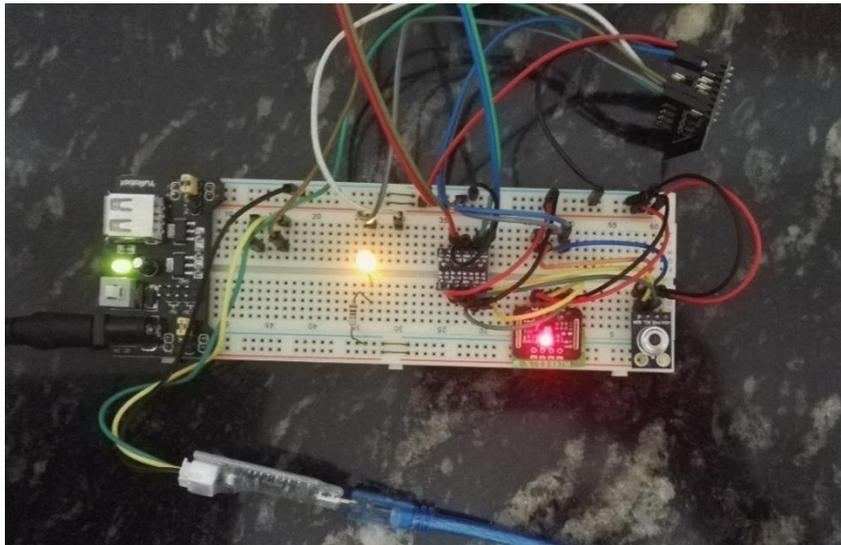


Figura 13 – Protótipo

### 5.3 Discussão entre o que foi proposto e o que foi feito

Creio que a grande dificuldade encontrada foi o desenvolvimento do firmware para o módulo sensor. As informações da fabricante do nRF52832, a Nordic Semiconductor e de seu SDK muitas vezes são insatisfatórias. Despendi muito tempo testando as APIs e procurando soluções para erros internos do SDK.

Como o estudo do SDK consumiu muito tempo, apenas o desenvolvimento de um protótipo de módulo sensor foi concluído no período de Projeto Final I.

Para Projeto Final II ficariam então as seguintes etapas:

- Desenvolvimento:
  - Desenvolvimento do firmware para a estação base
  - Montagem do protótipo da estação base
  
- Pesquisa
  - Interpretação médica das leituras dos sensores
  - Discussão sobre a segurança dos dados transmitidos
  
- Testes:
  - Integração entre módulo sensor e estação base
  - Obtenção dos dados transmitidos pela estação base pela internet
  - Capacidade de transmissão de dados
  - Consumo de energia do módulo sensor
  - Confiabilidade dos dados aferidos

## 6. Conclusão

O progresso do projeto durante Projeto Final I foi bastante proveitoso. Trabalhar com o desenvolvimento de firmware para microcontroladores se mostrou uma atividade excelente para pôr em prática todo o conhecimento teórico e prático que adquiri durante o curso de Engenharia da Computação.

Consegui compreender o protocolo de comunicação I<sup>2</sup>C e desenvolver um software para a comunicação entre sensores e um microcontrolador. Essa tarefa se mostrou especialmente difícil pois o SDK do microcontrolador é mal documentado, requerindo que eu fizesse diversos testes para validar minhas teorias.

Além de software, durante o Projeto Final I tive a oportunidade de mexer com hardware. Trabalhar com protoboards e componentes eletrônicos deu um novo sentido ao conteúdo estudado na cadeira de Circuitos. Principalmente ao fazer um hardware específico reagir a um software, em baixo nível.

Embora não tenha conseguido concluir todo o proposto no plano de ação para esse primeiro momento, as atividades atrasadas foram reorganizadas para serem realizadas durante o Projeto Final II sem maior prejuízo ao planejamento.

## 7. Referências bibliográficas

[1] **nRF5 SDK Documentation**. Disponível em: [https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct\\_nrf52%2Fstruct%2Fnrf52.html&cp=4/](https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct_nrf52%2Fstruct%2Fnrf52.html&cp=4/) Acesso em: 30 set. 2020.

[2] **nRF52832 Product Specification v1.4**. Disponível em: [https://infocenter.nordicsemi.com/pdf/nRF52832\\_PS\\_v1.4.pdf](https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf) Acesso em: 30 set. 2020.

[3] **MAX30102 Datasheet**. Disponível em: <https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf> Acesso em: 30 set. 2020.

[4] **MLX90614 Datasheet**. Disponível em: <https://www.melexis.com/-/media/files/documents/datasheets/mlx90614-datasheet-melexis.pdf> Download em: 02 out. 2020.

[5] **ESP8266 Technical Reference**. Disponível em: [https://www.espressif.com/sites/default/files/documentation/esp8266-technical\\_reference\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf) Acesso em: 02 out. 2020.

[6] **NodeMCU Documentation**. Disponível em: <https://nodemcu.readthedocs.io/en/release/> Acesso em: 03 out. 2020.

[7] **MQTT v3.1.1 Specification**. Disponível em: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf> Acesso em: 09 out. 2020.

[8] **I<sup>2</sup>C-bus Specification**. Disponível em: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf> Acesso em: 25 nov. 2020.

[9] **Undesrstanding the I<sup>2</sup>C-bus**. Disponível em: <https://www.ti.com/lit/an/slva704/slva704.pdf> Acesso em: 25 nov. 2020.

[10] **Adafruit MLX90614 Library**. Disponível em: <https://github.com/adafruit/Adafruit-MLX90614-Library> Acesso em: 25 nov. 2020.

[11] **SparkFun MAX3010x Sensor Library**. Disponível em:  
[https://github.com/sparkfun/SparkFun\\_MAX3010x\\_Sensor\\_Library](https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library) Acesso em:  
25 nov. 2020.