



**UNIVERSIDADE DO SUL DE SANTA CATARINA**

**RICARDO JÚNIOR ABREU DA SILVA**

**RODRIGO MENDES VIEIRA**

**MONITORAMENTO BASEADO EM SENSORES PARA IDENTIFICAÇÃO DE  
CONTAMINAÇÃO DE ÁGUA**

Palhoça

2020

**RICARDO JÚNIOR ABREU DA SILVA**  
**RODRIGO MENDES VIEIRA**

**MONITORAMENTO BASEADO EM SENSORES PARA IDENTIFICAÇÃO DE  
CONTAMINAÇÃO DE ÁGUA**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistema de Informação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Sistema de Informação.

Orientador: Prof. Flávio Ceci, Dr.

Palhoça  
2020

**RICARDO JÚNIOR ABREU DA SILVA**

**RODRIGO MENDES VIEIRA**

**MONITORAMENTO BASEADO EM SENSORES PARA IDENTIFICAÇÃO DE  
CONTAMINAÇÃO DE ÁGUA**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Sistema de Informação e aprovado em sua forma final pelo Curso de Graduação em Sistema de Informação da Universidade do Sul de Santa Catarina.

Palhoça, 14 de Julho de 2020.

---

Professor e orientador Flávio Ceci, Dr.  
Universidade do Sul de Santa Catarina

---

Professora Dra. Anelise Leal Vieira Cubas.  
Universidade do Sul de Santa Catarina

---

Professor Richard Henrique de Souza, M.Sc.  
Universidade do Sul de Santa Catarina

Dedico este trabalho, bem como minha trajetória acadêmica, aos meus pais, José Ricardo e Maria da Graça, minhas irmãs Heloiza e Marita, que durante toda minha caminhada me apoiaram e incentivaram.

**Ricardo Júnior Abreu da Silva**

Dedico este trabalho, bem como minha trajetória acadêmica, aos meus pais, Hilário(in memoriam) e Lúcia, minhas irmãs Joice e Mayara, que durante toda minha caminhada me apoiaram e incentivaram.

**Rodrigo Mendes Vieira**



## **AGRADECIMENTOS**

**Ricardo Júnior Abreu da Silva** agradece:

Primeiramente agradeço a Deus por ser a luz da minha vida, por me proporcionar competência para seguir em frente sempre, guiando meus passos em comunhão a ele.

Aos meus pais, José Ricardo e Maria da Graça, meu porto seguro, pilares fundamentais para a minha vida, que a todo o momento estão ao meu lado me apoiando e incentivando com muita sabedoria e amor.

A minhas irmãs Heloiza e Marita, por também estarem sempre ao meu lado, me ajudando a conduzir tudo, com confiança e perseverança.

Aos mestres que contribuíram com a minha trajetória acadêmica e profissional, tornando possível a graça desta formação.

Ao meu grande amigo Rodrigo Mendes Vieira, que leva com lealdade nossa amizade e companheirismo, proporcionando comigo o desenvolvimento deste trabalho.

Ao nosso orientador, Flávio Ceci, pela sua dedicação, clareza e paciência durante o desenvolvimento deste trabalho, proporcionando qualidade e responsabilidade a tudo que foi desenvolvido.

A professora Anelise Leal Vieira Cubas, responsável pelo laboratório de plasma, que nos proporcionou a oportunidade para o desenvolvimento deste projeto, com toda dedicação, apoio e sabedoria, tornando possível a eficiência do projeto.

A professora Maria Inés Castiñeira, por todo o apoio e suporte necessários durante as matérias de TCC.

À Universidade do Sul do Estado da Santa Catarina, Unisul, e a todos os professores que proporcionaram um ensino de qualidade e responsabilidade.

**Rodrigo Mendes Vieira** agradece:

Agradeço primeiramente a Deus por guiar meus passos e me dar força para chegar nesse momento.

A todos os mestres que contribuíram com a minha formação acadêmica e profissional durante a minha vida.

Aos meus pais Lúcia e Hilário (in memoriam) que sempre me incentivaram e me fizeram ter coragem para voltar a estudar, e me apoiaram em todas as áreas da minha vida, com seu amor e dedicação.

Aos meus tios Marlene e Juscelino (in memoriam) que são como pais para mim e sempre estiveram ao meu lado.

As minhas irmãs Joice e Mayara por estarem ao meu lado e por fazerem te confiança nas minhas decisões.

À Universidade do Sul do Estado de Santa Catarina, Unisul, e todos os seus professores que sempre proporcionaram um ensino de alta qualidade.

Ao meu orientador Flávio Ceci, pela sua dedicação e paciência durante o projeto. Seus conhecimentos fizeram grande diferença no resultado final deste trabalho.

“Seu trabalho vai ocupar uma grande parte da sua vida, e a única maneira de estar verdadeiramente satisfeito é fazendo aquilo que você acredita ser um ótimo trabalho. E a única maneira de fazer um ótimo trabalho é fazendo o que você ama fazer. Se você ainda não encontrou, continue procurando.” (Steve Jobs).



## RESUMO

A intensificação da atividade mineradora e seus processos, expandiram a economia nacional, proporcionando um complexo dinamismo econômico, causando profundas marcas nos ecossistemas regionais, bem como baixos índices de qualidade de vida e qualidade ambiental na região Sul do Estado de Santa Catarina. A poluição dos mananciais hídricos, é o mais grave impacto causado pela extração do carvão mineral, por estarem próximos a jazidas onde o mineral é explorado, decorrente de efluentes ácidos provenientes das minas e dos rejeitos de carvão. Afim de contribuir com o processo de tratamento da água contaminada com os efluentes da extração do carvão mineral, o laboratório de plasma da Unisul desenvolveu um reator, que utiliza a tecnologia do plasma frio por descarga corona, responsável pela ionização de um gás a baixa temperatura, do qual possui a vantagem de promover a ocorrência simultânea do processo de remediação dos efluentes de mineração, por completo, estando a frente dos processos químicos de custos elevados, via adição de agentes alcalinos, e/ou sulfetos para precipitar os metais dissolvidos na forma de hidóxidos, carbonatos e sulfetos. Com o intuito de contribuir com os processos realizados no laboratório, bem como resolver a problemática relacionada a novas contaminações possíveis durante o processo de remediação com o reator de plasma, o desenvolvimento de um sistema embarcado, capaz de coletar dados em tempo real e em simultâneo ao processo de remediação dos efluentes foi posto em prática. O sistema é composto por sensores que coletam pH, temperatura, umidade, pressão e frequência acoplados a um Becker com água contaminada, além de uma interface gráfica que apresenta esses dados no momento da coleta, bem como funções para salvar, buscar e exportar resultados obtidos. Para o desenvolvimento do projeto, referindo-se ao módulo embarcado, foi utilizado a plataforma de desenvolvimento BeagleBone Black, sensores físicos e a linguagem de programação C. Para interface gráfica, desenvolvida como web, foi utilizado o framework VUE, bem como componentes JavaScript e outras ferramentas para auxiliar no desenvolvimento e depuração da aplicação. Para armazenamento dos dados, optou-se por utilizar o PostgreSQL. Com o protótipo desenvolvido, foram realizados testes, bem como a avaliação dos resultados obtidos, no laboratório, acompanhados pela professora responsável Anelise Leal Vieira Cubas, e comprovado que a possibilidade de obter-se os dados em tempo real, minimiza alterações nos dados causados pela instabilidade nos processos do plasma, bem como a exatidão dos dados obtidos simultaneamente, ou seja, coletar em conjunto valores de

pH, temperatura, umidade, pressão e frequência. Levando em consideração estes benefícios, garantiu-se que água em tratamento não seja contaminada com outras bactérias.

Palavras-chave: Plasma. Embarcado. Sensor.

## ABSTRACT

The intensification of mining activity and its processes, expanded the national economy, providing a complex economic dynamism, causing deep marks in regional ecosystems, as well as low levels of quality of life and environmental quality in the southern region of the State of Santa Catarina. The pollution of water sources is the most serious impact caused by the extraction of mineral coal, as they are close to deposits where the mineral is explored, due to acidic effluents from mines and coal tailings. In order to contribute to the process of purifying water contaminated with the effluents from the extraction of mineral coal, the Unisul plasma laboratory developed a reactor, which uses the technology of cold plasma by corona discharge, responsible for the ionization of a gas at low temperature, which has the advantage of promoting the simultaneous occurrence of the mining effluent remediation process, completely, being at the forefront of high cost chemical processes, via the addition of alkaline agents, and / or sulfides to precipitate dissolved metals in the form of hydroxides, carbonates and sulfides. In order to contribute to the processes carried out in the laboratory, as well as to solve the problem related to possible new contamination during the process of remediation with the plasma reactor, the development of an embedded system, capable of collecting data in real time and simultaneously the effluent remediation process was put in place. The system consists of sensors that collect pH, temperature, humidity, pressure and frequency coupled to a Becker with contaminated water, in addition to a graphical interface that presents this data at the time of collection, as well as functions to save, search and export results. For the development of the project, referring to the embedded module, the BeagleBone Black development platform, physical sensors and the C programming language were used. For the graphical interface, developed as a web, the VUE framework was used, as well as JavaScript and other tools to assist in the development and debugging of the application. For data storage, it was decided to use PostgreSQL. With the prototype developed, the tests were carried out, as well as the evaluation of the results obtained, in the laboratory, accompanied by the responsible teacher Anelise Leal Vieira Cubas, and the possibility of obtaining the data in real time was verified, minimizing the changes in the data used by the instability in plasma processes, as well as the accuracy of the data used simultaneously, that is, jointly collected values of pH, temperature, temperature, pressure and frequency. Taking these benefits into account, we ensure that the water in the treatment is not contaminated by other bacteria.

Keywords: Plasma. On board. Sensor.





## LISTA DE ILUSTRAÇÕES

Figura 1 - Representação de um microprocessador AVR em blocos .....	30
Figura 2 - Indicação da relação entre resistência elétrica e deslocamento .....	33
Figura 3 - Acoplamento de um potenciômetro linear à haste de um cilindro pneumático.....	34
Figura 4 - Funcionamento de uma microchave .....	34
Figura 5 - Etapas metodológicas .....	37
Figura 6 - Arquitetura da solução .....	39
Figura 7 – Diagrama de casos de uso do sistema .....	45
Figura 8 – Diagrama de atividades do sistema embarcado .....	48
Figura 9 – Diagrama UML do back-end .....	49
Figura 10 – Protótipo de tela para acessar o sistema.....	50
Figura 11 – Protótipo de tela para apresentar dados em tempo real.....	51
Figura 12 – Protótipo de tela para área administrativa de exportação de relatórios.....	51
Figura 13 – Diagrama de estados finitos do sistema .....	55
Figura 14 – O hardware BeagleBone Black .....	57
Figura 15 – Fonte de alimentação.....	59
Figura 16 – GPIOs BeagleBone Black.....	60
Figura 17 – phMetro .....	61
Figura 18 – Sensores de umidade e temperatura .....	62
Figura 19 – Frequencímetro .....	63
Figura 20 – Barômetro.....	64
Figura 21 – Tampa para o becker .....	65
Figura 22 – Esquema de ligações .....	66
Figura 23 – Modelo conceitual de dados do sistema.....	67
Figura 24 – Modelo entidade relacional do sistema .....	68
Figura 25 – Componentes de máquina de estados.....	70/71
Figura 26 – Máquina de estados do sistema .....	72
Figura 27 – Funções para captura de dados.....	73
Figura 28 – Funções para cálculos de amostragem .....	75
Figura 29 – Início da coleta .....	83
Figura 30 – Coleta intermediária I.....	84

Figura 31 – Coleta intermediária II .....	85
Figura 32 – Fim da coleta .....	85
Figura 33 – Tela de login.....	86
Figura 34 – Tela de coleta de dados .....	87
Figura 35 – Tela de registro de coletas.....	87
Figura 36 – Tela de busca de coletas de dados.....	88
Figura 37 – JavaScript/Cadastro de usuários.....	89
Figura 38 – Node.js / Arquivo de configurações de dependências .....	90
Figura 39 – Express .....	90
Figura 40 – PostgreSQL / Coleta de dados.....	91
Figura 41 – Knex / Querys para dados de usuário.....	91
Figura 42 – Vue Router / Rotas da interface .....	92
Figura 43 – Vuetify / Botões Play, Pause e Stop.....	93
Figura 44 – Axios .....	93
Figura 45 – Postman .....	94
Figura 46 – Vuetools .....	95
Figura 47 – Webtools .....	95



## LISTA DE TABELAS

Quadro 1 – Matriz de rastreabilidade UC x RF .....	46
Quadro 2 – Estados do Sistema .....	52
Quadro 3 – Início da coleta.....	83
Quadro 4 – Coleta intermediária I .....	84
Quadro 5 – Coleta intermediária II.....	84
Quadro 6 – Fim da coleta .....	85
Quadro 7 – Investimento do protótipo.....	98

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>21</b>
1.1. PROBLEMÁTICA.....	22
1.2. OBJETIVOS.....	23
<b>1.2.1. Objetivo Geral.....</b>	<b>23</b>
<b>1.2.2. Objetivos Específicos .....</b>	<b>23</b>
1.3. JUSTIFICATIVA.....	24
1.4. ESTRUTURA DA MONOGRAFIA.....	24
<b>2. REFERENCIAL TEÓRICO .....</b>	<b>26</b>
2.1. POLUIÇÃO DA ÁGUA .....	26
2.2. A TECNOLOGIA DO PLASMA FRIO .....	27
2.3. OS SISTEMAS EMBARCADOS.....	29
2.4. OS SENSORES ELETRÔNICOS.....	31
<b>2.4.1. Variáveis .....</b>	<b>32</b>
<b>2.4.2. Atuadores.....</b>	<b>32</b>
2.5. SENSORES ANALÓGICOS.....	33
2.6. SENSORES DIGITAIS.....	34
2.7. MÁQUINAS DE ESTADOS FINITOS.....	35
<b>3. METODOLOGIA DE PESQUISA .....</b>	<b>36</b>
3.1. CARACTERIZAÇÃO DO TIPO DE PESQUISA .....	36
3.2. ETAPAS METODOLÓGICAS .....	37
3.3. PROPOSTA DE SOLUÇÃO .....	39
3.4. DELIMITAÇÕES .....	40
<b>4. PROJETO DE SOLUÇÃO .....</b>	<b>41</b>
4.1. REQUISITOS.....	41
<b>4.1.1. Requisitos do funcionais .....</b>	<b>42</b>
<b>4.1.2. Requisitos não funcionais .....</b>	<b>42</b>
<b>4.1.3. Requisitos funcionais e requisitos não funcionais do sistema .....</b>	<b>42</b>
4.2. DIAGRAMA DE CASOS DE USO .....	43
<b>4.2.1. Diagrama de caso de uso do sistema .....</b>	<b>44</b>
<b>4.2.2. Matriz de rastreabilidade.....</b>	<b>46</b>
4.3. DIAGRAMA DE ATIVIDADES .....	47
<b>4.3.1. Diagrama de atividades do sistema embarcado .....</b>	<b>48</b>

4.4. MODELAGEM UML .....	49
4.5. A APLICAÇÃO WEB .....	50
4.6. O SOFTWARE EMBARCADO .....	52
4.6.1. Estados do sistema .....	52
4.6.2. Diagrama de estados finitos do sistema .....	55
4.7. O HARDWARE .....	56
4.7.1. O hardware BeagleBone Black .....	56
4.7.2. Representação física da BeagleBone Black.....	57
4.7.3. Representação gráfica das portas digitais e analógicas da BeagleBone Black.....	60
4.7.4. phMetro .....	61
4.7.5. Sensor de umidade e temperatura.....	62
4.7.6. Frequencímetro .....	63
4.7.7. Barômetro.....	63
4.7.8. Acoplamento dos sensores no becker .....	64
4.7.9. Esquema de ligações entre a BeagleBone Black e os Sensores.....	66
4.8. O BANCO DE DADOS .....	67
4.8.1. Modelo conceitual de dados .....	67
4.8.2. Modelo entidade relacionamento.....	68
5. FERRAMENTAS E RESULTADOS.....	69
5.1. DESENVOLVIMENTO .....	69
5.2. DESENVOLVIMENTO DO SISTEMA EMBARCADO .....	70
5.2.1. Componentes utilizados na máquina de estados do sistema .....	71
5.2.2. Representação da máquina de estados finitos do sistema desenvolvido em linguagem C .....	72
5.2.3. Funções para captura de dados dos sensores .....	73
5.2.4. Funções para cálculo final de amostragem.....	74
5.3. FERRAMENTAS DE APOIO .....	76
5.3.1. JavaScript.....	76
5.3.2. Node.js.....	76
5.3.3. Express .....	77
5.3.4. PostgreSQL.....	77
5.3.5. Knex .....	77
5.3.6. Vue.js.....	78
5.3.7. Vue router.....	78
5.3.8. Vuex.....	79
5.3.9. Vuetify .....	79
5.3.10. Axios .....	79
5.3.11. VSCODE.....	80

5.3.12. Postamn.....	80
5.3.13. Chrome.....	81
5.3.14. Chrome devtools.....	81
5.3.15. Vue devtools.....	81
5.3.16. Eclipse .....	82
5.3.17. Linguagem C .....	82
5.4. RESULTAODS OBTIDOS UTILIZANDO O SISTEMA .....	83
5.5. APLICAÇÃO WEB .....	86
5.6. APLICAÇÃO DAS FERRAMENTAS DE APOIO .....	88
5.7. AVALIAÇÃO DO PROJETO .....	96
5.8. CUSTO PARA O DESENVOLVIMENTO DO HARDWARE .....	98
<b>6. CONCLUSÃO.....</b>	<b>100</b>
6.1. CONCLUSÕES.....	100
6.2. TRABALHOS FUTUROS.....	101
<b>REFERÊNCIAS.....</b>	<b>103</b>
<b>APÊNDICES .....</b>	<b>108</b>
<b>APÊNDICE A – CRONOGRAMA I.....</b>	<b>108</b>
<b>APÊNDICE B – AUTORIZAÇÃO PARA USO DE NOME E RESPOSTAS DE AVALIAÇÃO DO PROJETO.....</b>	<b>108</b>



## 1. INTRODUÇÃO

A atividade mineradora e seus processos foi intensificada nos últimos anos pela expansão da economia nacional, proporcionando um complexo dinamismo econômico, produzindo marcas profundas nos ecossistemas regionais e, também, baixos índices de qualidade de vida e qualidade ambiental na região sul de Estado de Santa Catarina (RODRIGUES, JULIANA, 2019).

Dentre os impactos ambientais relacionados com a mineração do carvão, o mais grave deles é a poluição dos mananciais hídricos das regiões próximas às jazidas onde o mineral é explorado (ORTIZ, L. et al., 2002). Esta poluição hídrica é decorrente da geração de efluentes ácidos provenientes das minas e dos rejeitos de carvão, a chamada água de drenagem ácida ou água de mina é responsável pela contaminação dos rios do entorno, com valores de pH na faixa entre 1,5 e 4,0. Esta elevada acidez da água de mina promove a dissolução de muitos metais tóxicos (Ferro, Alumínio, Zinco, Cobre, Cádmio, Astatato, Nióbio e Chumbo dentre outros) presentes em outros minerais, constituindo sistemas ambientalmente complexos que requerem etapas de tratamento químicas, físicas e biológicas para sua remediação (DAUBERT , BRENNAN, 2007).

O processo de plasma frio por descarga corona responsável pela ionização de um gás a baixa temperatura (CUBAS, et al. 2015), possui a vantagem de promover a ocorrência simultânea do processo de remediação dos efluentes de mineração de forma completa. Destarte, os tratamentos clássicos envolvem processos químicos via adição de agentes alcalinos e/ou sulfetos para precipitar os metais dissolvidos na forma de hidróxidos, carbonatos ou sulfetos. Porém, os custos do tratamento são elevados dada a grande quantidade de agentes precipitantes necessárias, gerando também grandes quantidades de lodo tóxico. Em contraste, o processo de plasma frio por descarga corona responsável pela ionização de um gás a baixa temperatura (CUBAS, et al. 2015), possui a vantagem de promover a ocorrência simultânea do processo de remediação dos efluentes de mineração de forma completa. Este processo, visa o tratamento de água de mina utilizando a tecnologia de plasma frio a pressão atmosférica. O controle para confirmar a eficiência do tratamento será através da análise de pH, alcalinidade, turbidez, ferro e sulfetos adaptada do Standard Methods for the Examination of Water and Wastewater 21a ed. 3500 B, 3- 58 (2005).

O processo de plasma frio é formado por um gás ionizado parcialmente, no qual a energia média dos elétrons é consideravelmente mais elevada do que a dos íons e moléculas de

gás. A energia produzida para gerar o plasma frio é muito pequena sendo proporcional ao aumento da temperatura. A descarga é formada através da aplicação de um campo elétrico intenso, o que provoca a formação de auto-propagação eletrônica dentro do volume de gás (FRIEDMAN et al., 2007). Uma vez gerado o gás ionizado, os elétrons colidem com as moléculas, criando espécies quimicamente ativas conhecidas como radicais. Os radicais uma vez produzidos podem substituir a química convencional em vários processos, no caso da água de mina, reduzem o sulfato para ácido sulfídrico, precipitando os íons metálicos dissolvidos em forma de sulfetos. Porém, durante o processo de tratamento, há grandes chances da água em experimento sofrer novas contaminações por conta da necessidade de realizar aferimentos em determinados intervalos de tempo, com equipamentos como o pHmetro e um termostato, por exemplo, ao serem inseridos e retirados no recipiente que contém a água contaminada.

## 1.1. PROBLEMÁTICA

Com base nos resultados a serem obtidos através do processo de tratamento da água de mina com a aplicação do plasma frio, foi observado a necessidade de realizar o monitoramento de alguns dados durante o processo e em tempo real. Os dados em questão são: pH, temperatura, umidade, pressão e frequência. Atualmente, no Laboratório de Plasma de Unisul esses dados são capturados de forma manual, e com pouca precisão, por conta da necessidade de parar o reator de plasma durante o processo de tratamento, para que sensores multi-parâmetros sejam inseridos no becker que contém a água em tratamento. Porém, ao realizar o processo de aferimento dos dados desta forma, resultados como o pH, por exemplo, podem ser muito diferentes em comparação aos valores obtidos enquanto o reator está em funcionamento. Há também grandes chances de contaminar a água em tratamento com outras bactérias que estejam fora do becker, apresentando resultados totalmente diferentes do esperado, e possivelmente necessitando a coleta de novas quantidades de água para novos testes (CUBAS, et al. 2015). Esses fatos, são totalmente comprometedores ao trabalho realizado em laboratório, comprometendo tempo, mão de obra, investimento de outros materiais necessários durante a realização destes experimentos. Tendo em vista a necessidade de otimizar o monitoramento do processo de tratamento da água contaminada, formulou-se a seguinte pergunta: Um sistema

composto de sensores, capaz de monitorar em tempo real o processo de tratamento da água, irá colaborar com a precisão e o resultado esperado utilizando a tecnologia do plasma frio?

## 1.2. OBJETIVOS

Esta seção apresenta os objetivos geral e específicos do presente trabalho.

### 1.2.1. Objetivo Geral

Desenvolver uma solução para monitoramento do tratamento de água de mina por Plasma frio baseado no uso de Sensores.

### 1.2.2. Objetivos Específicos

- Selecionar sensores para aplicar no processo de tratamento de água minha por plasma frio.
- Desenhar uma proposta de solução baseada em software e hardware.
- Desenvolver um protótipo funcional a partir da proposta de solução.
- Avaliar o protótipo desenvolvido em um laboratório.

### 1.3. JUSTIFICATIVA

O Laboratório de plasma da Unisul, é um ambiente rico em pesquisas voltadas a preservação do meio ambiente, bem como o desenvolvimento de recursos economicamente viáveis que contribuem para que esse processo aconteça.

Os trabalhos executados em laboratório contam com uma equipe de pesquisadores, professores e engenheiros, com metas claras e objetivas, em prol de obter os melhores e mais otimizados resultados.

Tendo a vista a necessidade de realizar o monitoramento do processo de tratamento de água de minha através do plasma frio (CUBAS, et al. 2015), sendo um dos principais trabalhos do laboratório, deu-se início ao desenvolvimento de um sistema de monitoramento, capaz de obter os dados necessários em tempo real, sem a necessidade de parar a aplicação do plasma. Desta forma, os resultados obtidos tornam-se mais precisos e confiáveis, prevenindo a água em tratamento de uma nova contaminação, e contribuindo para a eficiência dos experimentos.

Além de realizar o monitoramento em tempo real, será possível armazenar os dados obtidos em um banco de dados, que possibilitarão comparações entre experimentos realizados ao longo do tempo. Esses dados poderão ser exportados do sistema em formato de planilha, possibilitando o manuseio através do Microsoft Excel.

O sistema possibilitará também o monitoramento dos parâmetros de forma independente, ou seja, caso necessário monitorar apenas o pH, os demais sensores podem ser desabilitados antes de iniciar o monitoramento. Sendo assim, é possível observar que o sistema traz grandes benefícios ao laboratório e a equipe, comprovando a precisão dos resultados, o armazenamento dos dados obtidos, e garantindo qualidade ao processo de tratamento de água, que por sua vez será protegida de novos efluentes ou outros tipos de contaminação.

### 1.4. ESTRUTURA DA MONOGRAFIA

Este trabalho está dividido de forma em que o capítulo um se refere à introdução, que aborda a atividade mineradora no estado de Santa Catarina, com foco na extração de carvão e seus impactos ambientais. Bem como a formulação da problemática de pesquisa, os objetivos

propostos e a justificativa para realização deste trabalho. Abrangendo o propósito de monitorar o processo de tratamento da água contaminada utilizando a tecnologia do plasma frio.

O capítulo dois apresenta um foco na fundamentação teórica, o qual tem a finalidade de construir uma base para a formulação da pesquisa, onde são abordados os reais fatos sobre a poluição da água por meio da extração de carvão em Santa Catarina, a tecnologia e aplicação do plasma frio, bem como o conceito geral sobre os sistemas embarcados, e de que forma ele contribui com a solução desenvolvida.

No capítulo três, é apresentado o método de pesquisa utilizado para o desenvolvimento do presente trabalho.

O capítulo quatro apresenta as metodologias utilizadas para o desenvolvimento do sistema, incluindo a aplicação web, o sistema embarcado, bem como a modelagem do protótipo da solução e as definições e características de hardware.

O capítulo cinco apresenta as ferramentas utilizadas para dar suporte ao protótipo da solução, partes do desenvolvimento da aplicação web e do sistema embarcado, bem como o desenho da solução e avaliação dos resultados encontrados.

Por fim, o capítulo seis apresenta as conclusões e os trabalhos futuros.

## 2. REFERENCIAL TEÓRICO

O presente capítulo tem o objetivo de apresentar uma revisão teórica dos reais fatos sobre a poluição da água, a tecnologia e aplicação do plasma frio, bem como um conceito geral sobre sistemas embarcados.

Será apresentado também uma abordagem sobre os sensores eletrônicos que contribuem para o desenvolvimento de soluções para as mais diversas áreas de mercado.

### 2.1. POLUIÇÃO DA ÁGUA

A mineração de carvão é uma importante atividade econômica da região sul de Santa Catarina. No entanto, a mesma tem gerado sérios problemas ambientais em virtude do mau gerenciamento de seus rejeitos, que podem contaminar o solo e meio hídrico local (CAMPOS; ALMEIDA; SOUZA, 2003). Mesmo após o término das operações de extração e beneficiamento do carvão, a geração de poluentes continua ativa, sendo extremamente danosa ao meio ambiente por décadas, e alguns casos, centenas de anos (ALEXANDRE, 1996).

Dentre os contaminantes provenientes da atividade mineradora encontram-se os rejeitos piritosos que são os principais geradores das drenagens ácidas de mina (DAM) (AKCIL; KOLDAS, 2006), caracterizadas pela elevada acidez e expressiva concentração de metais pesados como cádmio, chumbo, cobre, ferro, alumínio, manganês, zinco e além de sulfatos (BENASSI et al., 2006; LAUS et al., 2006). A geração da DAM é decorrente da percolação da água de chuva através dos rejeitos gerados nas atividades de lavra e beneficiamento do carvão, que pode alcançar os corpos hídricos superficiais e/ou subterrâneos levando a degradação da qualidade de suas águas (BANKS; BANKS, 2001).

A DAM, quando responsável pela poluição hídrica, altera os níveis de pH das águas e libera metais pesados que possuem um elevado potencial de toxicidade (BELL; BULLOCK, 1996). Estes metais podem ser perigosos para populações animais e vegetais locais, pois podem se acumular nos sedimentos de rios (NIETO et al., 2007), serem absorvidos diretamente ou indiretamente através do processo de biomagnificação (CARDWELL; HAWKER;

GREENWAY, 2002) e comprometer a qualidade de vida de populações que vivem próximas a um corpo d'água contaminado (ALBERING et al., 1999).

O potencial tóxico da DAM já foi descrito por alguns trabalhos (BENASSI, 2004; BENASSI et al., 2006; GEREMIAS et al. 2006; GERHARDT et al., 2004). Também foi demonstrado que inundações em ambientes geradores de DAM, podem impactar outros ambientes aquáticos distantes até 25km do centro de inundação, se estes estiverem a sua jusante (LIN et al., 2007). Desta maneira a contaminação de ambientes aquáticos por DAM se apresenta como um problema ambiental de grande magnitude com impactos sobre áreas adjacentes e até mesmo distantes. Além disto, a remediação da contaminação ambiental por DAM é de difícil resolução (BANKS; BANKS, 2001), uma vez que muitos métodos para seu tratamento possuem custos elevados ou produzem materiais residuais (JOHNSON; HALLBERG, 2005; SHEORAN; SHEORAN, 2006).

Na região carbonífera de Santa Catarina, a poluição hídrica causada pela DAM é provavelmente um dos impactos mais significativos das operações de mineração e beneficiamento de carvão (ALEXANDRE; KREBS, 1995). Dentre os ambientes aquáticos na rede hidrográfica do sul catarinense que sofrem com a contaminação por rejeitos da mineração de carvão encontra-se o Rio Molha. As águas superficiais do Rio Molha são esporadicamente contaminadas por rejeitos provenientes de minas de carvão á céu aberto abandonadas, localizadas às suas margens, podendo comprometer a qualidade de suas águas.

## 2.2. A TECNOLOGIA DO PLASMA FRIO

Com base nos objetivos do desenvolvimento sustentável da agenda de 2030 estabelecida pela ONU, e como um dos desafios do milênio, destaca-se o acesso a água potável para a população mundial.

A reutilização e reciclagem provisionada pelo chamado “objetivo seis”, visa reduzir pela metade a proporção de água não tratada em todo o mundo. Com isso, torna cada vez mais importante e desafiador a pesquisa de novos métodos e processos capazes de realizar a remoção de contaminantes químicos e microbiológicos de água, pelo qual, alguns autores demonstram a utilização da tecnologia do plasma não térmico, e sua eficiência (MAGUREAU et al., 2011).

A aplicação do plasma não térmico para inativação de bactérias na água, é segura e econômica, não exige a adição de produtos químicos, como por exemplo o cloro, reagentes a matéria orgânica em meio aquoso.

Em princípio, a tecnologia do plasma com pressão atmosférica foi utilizada para o processo de esterilização, e recentemente alguns estudos demonstram resultados excelentes na inativação de *Escherichia coli* (*E. coli*) em água contaminada, sendo por descarga de plasma em superfície aquosa, ou por jato de plasma imerso ao meio líquido (THOPAN et al., 2017).

O plasma não térmico é um gás parcialmente ionizado, constituído por uma descarga elétrica de alta tensão, pela qual forma auto propagação de elétrons e espécies químicas ionizadas (EHLBECK et al., 2015).

A utilização do ar como gás de plasma, produz variadas espécies químicas instáveis, por conta do nitrogênio e o oxigênio sofrerem excitação, dissociação e ionização. O plasma de oxigênio produz Eqs. de ozônio, já o nitrogênio forma espécies excitadas, Eq., das quais contribuem para formar muitas espécies na água, como apresentado nas Eqs. (CHANDANA et al., 2018). O plasma de nitrogênio forma espécies de nitrito da água que reduzem o pH e reagem ainda mais com o peróxido de hidrogênio formado pelo plasma com a finalidade formar um forte peroxinitrito oxidante que pode danificar a membrana bacteriana e o DNA (BASHER et al., 2015).

Além das espécies apresentadas nas Eqs., acontece também a dissociação da água no plasma, como mostrado de forma resumida pelas Eqs., que levam a acidificação do meio a formação de espécies oxidantes, radicais como  $\cdot\text{H}$ ,  $\cdot\text{O}$ ,  $\cdot\text{OH}$  e peróxido de hidrogênio  $\text{H}_2\text{O}_2$  (LUKES et al., 2004). A dissociação e ionização acontece principalmente na interface de gás líquido (BENETOLI et al., 2012).

Uma vez formadas as espécies radicais químicas, conseguem substituir reações convencionais em variados processos, na água, por inativação de bactérias, as espécies plasmáticas causam oxidação e drenagem a membrana celular dos organismos. Além das espécies radicais, o ozônio produzido pelo plasma, conhecido por sua eficiência em esterilização (RAMDANI et al., 2015), sendo até 3.120 vezes mais letal para microorganismos quando comparado ao cloro. O plasma transmite também luz ultravioleta ( $h\nu$ ), conhecida como bactericida que produz espécies radicais (GORBANEV et al., 2016).

Portanto, a eficiência do reator de plasma não térmico utilizado para o tratamento de água depende de sua geometria e disposição dos eletrodos (BRUGGEMAN et al., 2016). As



descargas de alta tensão ocorrem entre eletrodos que podem ser organizados de diferentes maneiras dentro dos reatores de plasma, da qual reações químicas ocorrem na fase líquida, ou gasosa, com isso a massa e o calor são transferidos entre as fases.

### 2.3. OS SISTEMAS EMBARCADOS

Os sistemas embarcados tornaram-se parte do nosso dia-a-dia devido ao modo intenso pelo qual os circuitos digitais e as técnicas digitais passaram a ser utilizados em quase todas as áreas: computadores, automação, robôs, tecnologias e ciências médicas, transportes, telecomunicações, entretenimento, e assim por diante (TOCCI et al. 2011). Para chegarmos onde estamos hoje, as construções de sistemas digitais evoluíram desde o uso das grandes válvulas eletrônicas, passando pelos transístores até chegamos nos atuais circuitos integrados digitais.

Segundo Barros e Cavalcante (2010, p. 1):

O que pode surpreender, no entanto, é que bilhões de sistemas são produzidos anualmente para as mais diferentes propostas; tais sistemas estão embutidos em equipamentos eletrônicos maiores e executam repetidamente uma função específica de forma transparente para o usuário do equipamento.

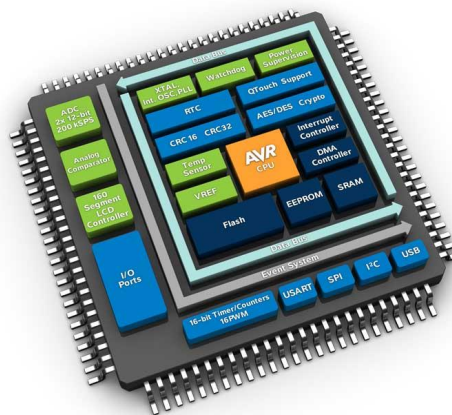
Essa evolução foi responsável por permitir a implementação de sistemas digitais cada vez mais complexos em um menor espaço físico, utilizando técnicas precisas e sofisticadas durante a fabricação dos circuitos integrados (MOORE 1998). O baixo custo de fabricação até mesmo dos complexos circuitos integrados (como o caso dos microprocessadores que usamos em nossos computadores, note-books e smartphones), permite que os projetistas incluam cada vez mais funcionalidades nestes sistemas, tornando-os adequados para as mais diversas aplicações.

Entretanto, esse baixo custo de fabricação está associado à produção em larga escala, o que inviabiliza o projeto e fabricação de alguns poucos circuitos integrados apenas para fins de teste, ou para utilização em aplicações muito específicas em pequena escala. Neste contexto, surgiram os Dispositivos Lógicos Programáveis (PLDs em inglês), que são componentes eletrônicos usados para construir circuitos digitais reconfiguráveis.

Diferente dos circuitos lógicos tradicionais, que possuem um funcionamento lógico fixo definido durante o projeto, os PLDs saem das fábricas sem nenhum comportamento fixo, podendo ser programados posteriormente pelos desenvolvedores de acordo com suas necessidades e se tornando propícios para o uso em casos de aprendizagem, prototipagem e testes em pequena escala.

O uso dos PLDs levou à criação de microcontroladores. Também conhecidos como MCU, são um computador num chip, contêm um processador, memória e periféricos de entrada e saída (E/S). O microcontrolador é como um microprocessador que pode ser programado para funções específicas, em contraste com outros microprocessadores de propósito geral, como os utilizados em computadores. Diferente dos PLDs, os microcontroladores não são apenas um circuito programável, mas um sistema completo que já oferece funcionalidades básicas como gerência de memória e acesso a dispositivos de E/S, conforme a Figura 1.

Figura 1 - Representação de um microprocessador AVR em blocos



Fonte: <https://start.atmel.com/>

Para facilitar ainda mais o desenvolvimento de sistemas utilizando microcontroladores, eles normalmente são acoplados à plataformas de prototipagem, como é o caso do Arduino (ARDUINO 2012) e do Mbed (HOLDINGS 2011).

Segundo Barros e Cavalcante (2010, p. 1):

Como resultado da introdução de sistemas eletrônicos em aplicações tradicionais temos produtos mais eficientes, de melhor qualidade e mais baratos. Dentre os componentes eletrônicos mais utilizados temos os componentes digitais que permitem algum tipo de computação tais como microprocessadores e microcontroladores. Apesar da melhoria na qualidade do produto final, o projeto de tais sistemas tornou-se bem mais complexo, principalmente por envolver uma série de

componentes distintos e de natureza heterogênea. O projeto de uma mesma aplicação pode incluir, por exemplo, transístores e instruções de máquina de um processador.

Essas plataformas oferecem não somente o microcontrolador, mas também outros componentes para realizar a interface com computadores (como portas USB, Ethernet, RS-232, etc.), componentes de microeletrônica básicos (LEDs e botões), além de outros dispositivos de E/S. Toda a programação destas plataformas é feita através de softwares que são instalados em computadores. Através dos ambientes desenvolvidos pelos próprios fabricantes, é possível programar todo o funcionamento desses microcontroladores, além de enviar o código já compilado para o dispositivo, normalmente através de uma interface USB.

## 2.4. OS SENSORES ELETRÔNICOS

Sensores eletrônicos são dispositivos sensíveis à alguma forma de energia do ambiente que pode ser luminosa, termica, cinética, relacionando informações sobre uma grandeza física que precisa ser mensurada (medida), como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc.

Conforme Wendling (2010, p. 3):

Sensores servem para informar um circuito eletrônico a respeito de um evento que ocorra externamente, sobre o qual ele deva atuar, ou a partir do qual ele deva comandar uma determinada ação.

Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Normalmente o sinal de saída deve ser manipulado antes da sua leitura no sistema de controle. Isso geralmente é realizado com um circuito de interface para produção de um sinal que possa ser lido pelo controlador. Supondo que a saída de um sensor, ao ser sensibilizado por uma energia externa, é dada por um nível de tensão muito baixo, torna-se necessária a sua amplificação (THOMAZINI et al., 2005).

Essa interface seria então um amplificador capaz de elevar o nível do sinal para sua efetiva utilização.

Há uma série de características relacionadas aos sensores que devem ser levadas em consideração na hora da seleção do sensor mais indicado para uma aplicação. No âmbito industrial, é fundamental uma grande quantidade de características, principalmente quando tratamos de automação e instrumentação industrial.

Inicialmente é necessário mostrar a diferenciação entre alguns elementos presentes em uma automação de qualquer natureza. Os principais elementos que atuam sobre a automação industrial são os sensores e atuadores, pois eles verificam e interferem no ambiente controlado (WENDLING, 2010).

#### **2.4.1. Variáveis**

São fenômenos físicos que chamamos simplesmente variáveis, por exemplo: temperatura, pressão, intensidade luminosa, etc. Cada sistema de medição pode ser compreendido em termos do que ele faz, por exemplo: indicar a temperatura ou totalizar a vazão ou registrar a pressão de um sistema qualquer.

#### **2.4.2. Atuadores**

São dispositivos que modificam uma variável controlada. Recebem um sinal proveniente do controlador e agem sobre o sistema controlado. Geralmente trabalham com potência elevada.

Exemplos de alguns atuadores:

- Válvulas (pneumáticas, hidráulicas);
- Relés;
- Cilindros (pneumáticos, hidráulicos);
- Motores;
- Solenóides;

- Etc.

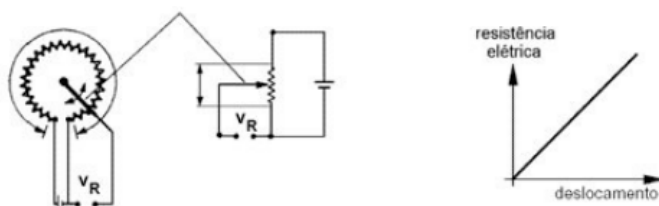
## 2.5. SENSORES ANALÓGICOS

Esse tipo de sensor pode assumir qualquer valor no seu sinal de saída ao longo do tempo, desde que esteja dentro da sua faixa de operação.

Essas variáveis são mensuradas por elementos sensíveis com circuitos eletrônicos não digitais, como por exemplo um potenciômetro de resistência, é um sensor que mede deslocamentos lineares ou angulares de acordo com a variação da resistência elétrica de um resistor.

A tensão elétrica de saída do potenciômetro, indicada por  $V_R$  na Figura 2, a seguir, é dada em função da posição de um terminal ou ponteiro sobre a resistência. Na Figura 2, o potenciômetro angular, é possível observar que o terminal está obtendo grande parte da resistência, ou seja, quanto mais no fim da resistência, maior é o deslocamento.

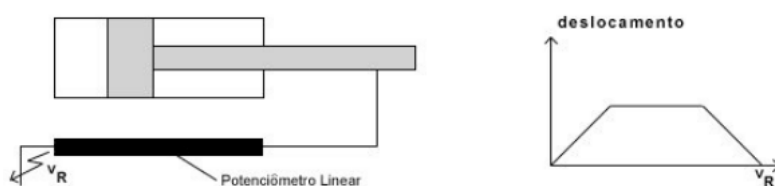
Figura 2 – Indicação da relação entre resistência elétrica e deslocamento



Fonte: THOMAZINI, 2005.

Conectando-se um potenciômetro linear à haste de um cilindro pneumático pode-se medir sua posição e seu deslocamento pelo mesmo princípio. Na Figura 3, onde o gráfico da direita mostra o deslocamento do cilindro pneumático em função da tensão elétrica de saída do potenciômetro.

Figura 3 - Acoplamento de um potenciômetro linear à haste de um cilindro pneumático



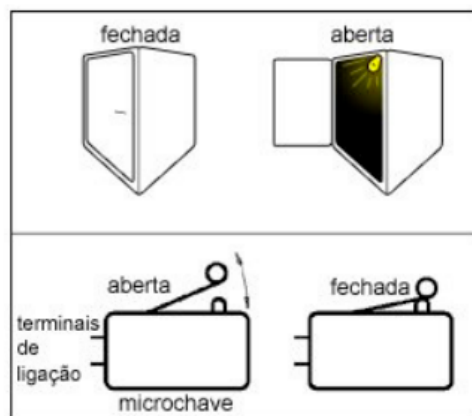
Fonte: THOMAZINI, 2005.

## 2.6. SENSORES DIGITAIS

Esse tipo de sensor pode assumir apenas dois valores no seu sinal de saída ao longo do tempo, que podem ser interpretados como zero ou um.

Não existem naturalmente grandezas físicas que assumam esses valores, mas eles são assim mostrados ao sistema de controle após serem convertidos por um circuito eletrônico (geralmente um comparador) (THOMAZINI et al., 2005), como por exemplo um detetor de contato (microchave ou sensor de fim de curso), geralmente utilizado em portas de geladeiras para acender e apagar a lâmpada interna. Na Figura 4, é possível observar o funcionamento deste tipo de sensor.

Figura 4 - Funcionamento de uma microchave



Fonte: THOMAZINI, 2005.

As microchaves indicam somente dois estados (ligado ou desligado, aberto ou fechado, etc.). Na indústria, este tipo de sensor pode ser usado para indicar se um cilindro pneumático atingiu ou não a posição correta, pode ser usado também para indicar o número de peças que passam por uma esteira. A chave é acionada a cada vez que uma peça passa e então envia um sinal para um contador ou para uma entrada digital de um CLP, que irá contar as peças. Ele faz a contagem baseado nos sinais da microchave. Se passarem 10 peças, 10 vezes a chave será acionada e mandará 10 vezes o mesmo sinal para o CLP.

## 2.7. MÁQUINAS DE ESTADOS FINITOS

As máquinas de estados finitos, são máquinas abstratas que capturam as partes essenciais de algumas máquinas concretas. Essas últimas vão desde máquinas de vender jornais e de vender refrigerantes, passando por relógios digitais e elevadores, até programas de computador, como alguns procedimentos de editores de textos e compiladores. O próprio computador digital, se considerarmos que sua memória é limitada, pode ser modelado por meio de uma máquina de estados finitos. Embora existam máquinas abstratas conceitualmente mais poderosas que as de estados finitos, que são mais adequadas para a modelagem de certas máquinas como o computador, as máquinas de estados finitos são adequadas, tanto do ponto de vista teórico quanto do prático, para a modelagem de amplo espectro de máquinas (mecânicas, eletrônicas, de software etc).

Existem basicamente dois tipos de máquinas de estados finitos: os transdutores e os reconhecedores ou aceitadores de linguagens. Os transdutores são máquinas com entradas e saídas, e os reconhecedores são máquinas com apenas duas saídas possíveis, ou seja, geralmente uma delas significa “aceitação” da entrada, e a outra, “rejeição” da entrada.

Uma característica fundamental de uma máquina de estados finitos é que sua memória é limitada e exclusivamente organizada em torno do conceito de “estado”. Embora uma máquina de estados finitos seja, tecnicamente falando, uma estrutura matemática, e um conjunto de estados um conjunto finito da estrutura, pode ser conveniente introduzir esses conceitos de maneira mais intuitiva do que normal (VIERIA, 2004).

### 3. METODOLOGIA DE PESQUISA

Este capítulo possui uma abordagem referente aos tipos de pesquisa aplicados neste trabalho, bem como as etapas de todos os processos necessários para atingir os objetivos propostos na solução inicial.

#### 3.1. CARACTERIZAÇÃO DO TIPO DE PESQUISA

Quanto a natureza de pesquisa, este trabalho é classificado como pesquisa bibliográfica. Segundo Maroni e Lakatos (2007, p.1), “a pesquisa bibliográfica, ou de fontes secundárias, abrange toda bibliografia já tornada pública em relação ao tema de estudo, desde publicações avulsas, boletins, jornais, revistas, livros, pesquisas, monografias, teses, material cartográfico e etc.”, bem como meios de comunicação orais, rádios, gravações em fita magnética e audiovisuais. Este tipo de pesquisa tem como finalidade posicionar o pesquisador em contato direto com tudo que foi escrito, dito ou filmado sobre determinado assunto, inclusive conferências seguidas de debates que tenham sido transcritos de alguma forma, quer publicadas ou regravadas. Em relação aos objetivos, a pesquisa do presente trabalho é classificada como qualitativa, por tratar-se de um método que difere, em princípio, do quantitativo, à medida que não emprega um instrumental estático como base na análise de um problema, não pretendendo medir ou numerar categorias (RICHARDSON, 1989).

Os estudos de campo qualitativos não têm um significado preciso em quaisquer das áreas onde sejam utilizados. Para alguns, todos os estudos de campo são necessariamente qualitativos e, mais ainda, identificam-se com a observação participante. É possível partir do princípio de que a pesquisa qualitativa é aquela que trabalha predominantemente com dados qualitativos, isto é, a informação coletada pelo pesquisador não é expressa em números, ou então números e suas conclusões neles baseadas representam um papel menor na análise (Dalfovo, et. al, 2008).

A classificação da pesquisa utilizada para este trabalho, em relação a abordagem, é a pesquisa qualitativa, pois os dados são obtidos a partir de sensores eletrônicos, diante da aplicação do plasma com a utilização da tecnologia do plasma frio, gerado através de um reator,



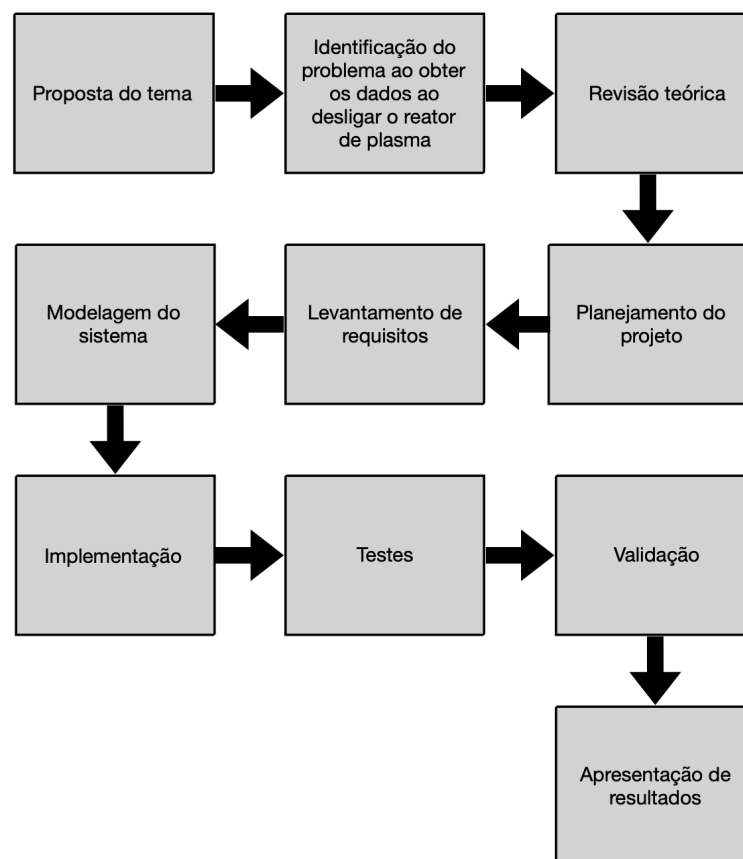
em um becker de vidro contendo água contaminada pela extração de carvão mineral. Com isso, é possível afirmar a não utilização de métodos estáticos, como é realizado em pesquisas quantitativas.

Por fim, com relação aos procedimentos técnicos utilizados para coleta de dados, e pesquisa deste trabalho é classificada como pesquisa bibliográfica, pois é considerado um dos métodos mais fáceis por se embasar em livros, artigos, pesquisas, meios eletrônicos e outros.

### 3.2. ETAPAS METODOLÓGICAS

A fim de atingir os objetivos propostos para este trabalho, ele foi fragmentado em 10 etapas metodológicas para facilitar o desenvolvimento da proposta inicial. Essas etapas podem ser observadas na Figura 5 e são detalhadas na sequência.

Figura 5 - Etapas metodológicas



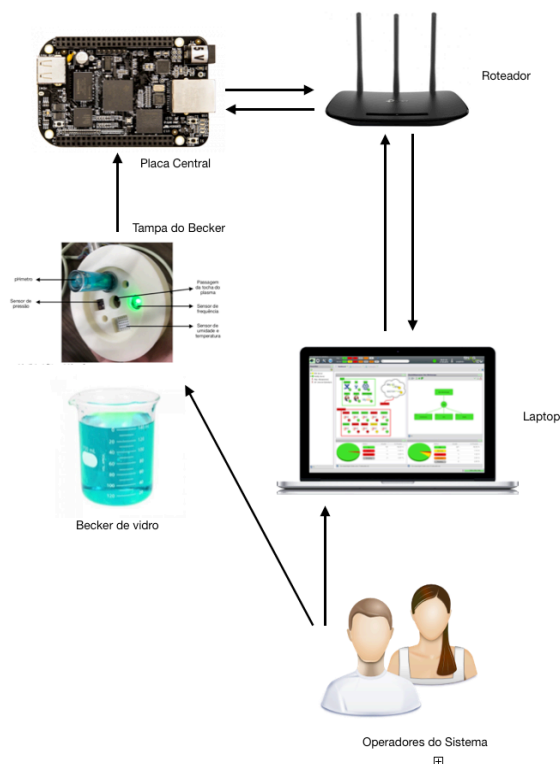
- **Proposta de tema:** Inicialmente, o Laboratório de Plasma da Unisul, que tinha a necessidade de mensurar alguns parâmetros em tempo real, durante o processo de tratamento da água contaminada com a extração de carvão mineral, sem desligar o reator de plasma.
- **Identificação do problema ao obter os dados ao desligar o reator de plasma:** Nesta etapa, são identificados os problemas ao mensurar os parâmetros da água em experimento ao desligar o reator de plasma e realizar o procedimento manualmente.
- **Revisão bibliográfica:** Para o êxito da proposta inicial, há a necessidade de realizar um levantamento teórico sobre o tema em diversas literaturas, para servir como base aos parâmetros necessários a serem obtidos, bem como os resultados obtidos com os métodos atuais.
- **Planejamento do projeto:** São identificados os métodos de pesquisas disposto no presente trabalho em relação aos seus objetivos, natureza, abordagem e procedimento utilizado para coleta de dados. Após, é realizado o planejamento de acordo com as informações coletadas durante a pesquisa.
- **Levantamento de requisitos:** Nesta etapa são identificados os requisitos funcionais e não funcionais, a fim de trazer praticidade durante a utilização do sistema ao realizar os experimentos, e acervo para consultas e comparações futuras.
- **Modelagem do sistema:** Para que seja possível uma implementação estruturada é necessária a modelagem do sistema. Com isso, será utilizada a técnica de diagramas UML. Serão formulados: protótipos de tela, casos de uso, diagrama de classe, modelagem do banco de dados e protocolo proprietário para comunicação.
- **Implementação:** Nesta fase, é desenvolvido o sistema com base na modelagem realizada, bem como com os conhecimentos adquiridos durante a realização deste trabalho.
- **Testes:** Com a finalidade de validar o sistema, são necessários a realização de testes para identificar possíveis erros na implementação ou até mesmo na modelagem realizada.

- **Validação:** Esta etapa tem o objetivo de validar o sistema, sendo disponibilizado ao Laboratório de plasma da Unisul, para que experimentos sejam realizados e identificar se os objetivos propostos foram alcançados.
- **Apresentação dos Resultados:** Os resultados obtidos com o trabalho proposto são analisados e apresentados, a fim de formular uma conclusão. Após a análise, será identificado se os objetivos propostos foram alcançados.

### 3.3. PROPOSTA DE SOLUÇÃO

Tendo em vista os objetivos pretendidos, a solução proposta será construída sobre uma demanda exclusiva do Laboratório de Plasma da Unisul. A solução pode ser definida como um equipamento que irá auxiliar o trabalho diário no laboratório, bem como, facilitar a busca de resultados de experimentos realizados com o sistema em momentos futuros. Na Figura 6, é possível identificar a arquitetura da solução.

Figura 6 - Arquitetura da solução



O acesso ao sistema pode ser realizado a partir de qualquer computador desktop ou laptop em rede local e dispensa a necessidade de internet, através de login e senha de usuários previamente cadastrados. Os dados cadastrados são armazenados em um banco de dados local, que terá comunicação com a aplicação web. Além do cadastro de usuários, também serão registrados relatórios referentes aos experimentos realizados, contendo informações de usuário, data, hora e resultados obtidos.

### 3.4. DELIMITAÇÕES

Devido ao desenvolvimento da solução ser voltado a uma necessidade do laboratório, são definidas algumas limitações para o sistema:

- A modelagem do banco de dados é única e exclusiva aos atributos a serem preenchidos conforme os requisitos do sistema;
- O software, bem como hardware foram desenvolvidos com compatibilidade exclusiva aos sensores em questão;
- Não é foco deste projeto especificar políticas de segurança da informação e de software;

## **4. PROJETO DE SOLUÇÃO**

Este capítulo aborda de forma geral os passos e técnicas utilizados para o desenvolvimento completo da solução.

A solução é composta por Software, Hardware e Banco de dados. O software, foi desenvolvido em duas partes, uma aplicação Web e um sistema embarcado. O hardware é composto pela plataforma de desenvolvimento BeagleBone Black e os sensores, sendo eles, para medir pH, temperatura, umidade, pressão e frequência.

### **4.1. REQUISITOS**

A fim de desenvolver qualquer sistema, a fase de definição dos requisitos e, conseqüentemente, a de especificação são etapas fundamentais do projeto. Os requisitos possuem uma função importante, pois o sucesso do sistema é dado pelo grau em que foram atendidos os requisitos e objetivos propostos durante o seu desenvolvimento (FALBO, 2013).

#### **4.1.1. Requisitos funcionais**

Requisitos funcionais referem-se sobre o que sistema deve fazer, ou seja, suas funções e informações. Com isso, requisitos funcionais são responsáveis pelas funcionalidade e os serviços do sistema, bem como, o que o sistema deve fornecer ao cliente e como o sistema irá se comportar em determinadas situações (SOMMERVILLE, 2003).

#### **4.1.2. Requisitos não funcionais**

Requisitos não funcionais são referidos aos critérios que qualificam os requisitos funcionais. Esses critérios podem ser de qualidade para o software, ou seja, requisitos de performance, usabilidade, confiabilidade, robustez entre outros. Os critérios também podem ser quanto a qualidade para o processo de software como requisitos de entrega e implementação (SOMMERVILLE, 2003).

#### **4.1.3. Requisitos funcionais e requisitos não funcionais do sistema**

Com base na necessidade abordada pelo laboratório de plasma de Unisul, do qual levou ao desenvolvimento deste projeto, foram levantados os requisitos abaixo, sendo eles funcionais e não funcionais.

##### **Requisitos funcionais:**

- RF001: Botão para iniciar e parar a coleta de dados;
- RF002: Botão para salvar os dados coletados;
- RF003: Botão para exportar os dados obtidos em formato PDF e/ou CSV;
- RF004: Botão para procurar resultados salvos anteriormente utilizando a data e/ou nome do responsável;
- RF005: Campo para inserir o nome do responsável pela coleta de dados;
- RF006: Inserir automaticamente ID, bem como, data e hora do início e fim da coleta de dados;
- RF007: Botão para excluir um resultado salvo;
- RF008: Botão para descartar um resultado que ainda não foi salvo;
- RF009: Campo para habilitar os sensores que deseja realizar a coleta de dados;
- RF010: Menu para apresentar resultados salvos;
- RF011: Menu para gerenciamento de resultados;
- RF012: Gerenciamento de coleta de dados;

**Requisitos não funcionais:**

- RNF001: Banco de dados para armazenar os resultados obtidos;
- RNF002: Plugin que possibilite exportar os resultados em formato PDF e CSV;
- RNF003: Utilizar um kit de desenvolvimento embarcado que possua sistema operacional Linux, compatível com servidores web, linguagem de programação C, sistemas de banco de dados, acesso a internet e espaço para armazenamento;
- RNF004: Sensores eletrônicos para obter os seguintes dados: pH, temperatura, umidade, frequência e pressão;

#### 4.2. DIAGRAMA DE CASO DE USO

O diagrama de caso de uso documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não são aprofundados dados técnicos que dizem como o sistema faz.

Este artefacto é comumente derivado da especificação de requisitos, que por sua vez não faz parte da UML (DEVMEDIA, 2012).

Diagramas de caso de uso, são compostos basicamente por quatro partes:

- Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema;
- Ator: Usuário do sistema, ou melhor, um tipo de usuário;
- Use Case: É uma tarefa ou uma funcionalidade realizada pelo ator (usuário);
- Comunicação: É o que liga um ator com um caso de uso;

#### **4.2.1. Diagrama de caso de uso do sistema**

O diagrama de caso de uso abaixo, foi construído tendo como base os requisitos funcionais apontados na seção 4.1.3.

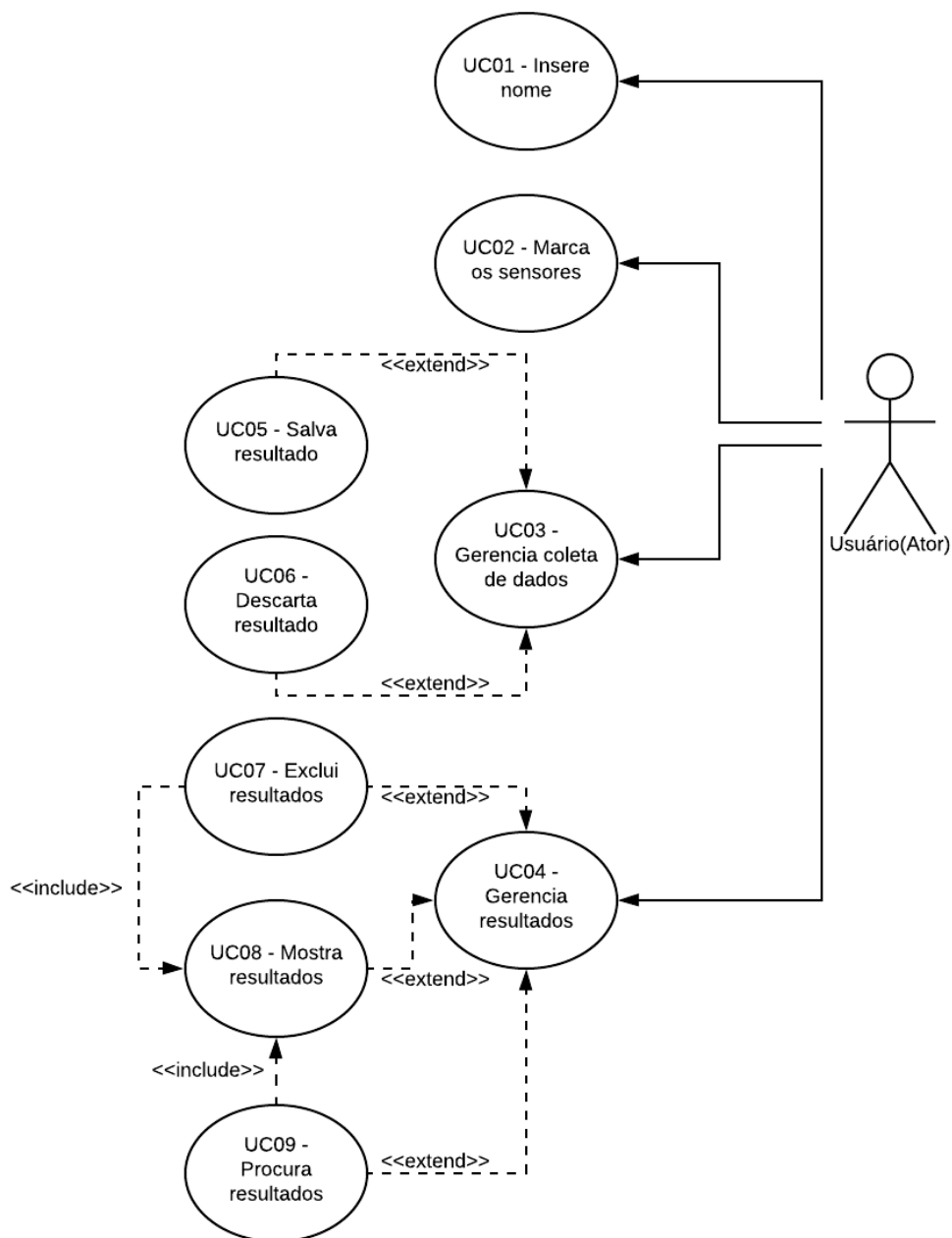
De acordo com as partes que compõe este diagrama, o ator é representado pelo usuário, ou seja, responsável pela coleta de dados, suas ações são representadas pelo cenário que as contém, exemplo, insere nome, e interligados com o usuário através de setas lisas.

Neste diagrama foi aplicado também o conceito  $< >$ , ou seja, podendo conter include e/ou extend, que são utilizados para representar as relações entre os casos.

Include, representa a relação de um caso de uso que para ter sua funcionalidade executada precisa chamar outro caso de uso. Já o Extend, significa que o caso de uso estendido vai funcionar exatamente como o caso de uso base, porém com alguns novos passos inseridos no caso de uso estendido. Neste diagrama, foi utilizado apenas o Include (DEV MEDIA, 2012).



Figura 7 – Diagrama de casos de uso do sistema



Fonte: os autores, 2020.

De acordo com a Figura 7, os casos de uso interligados via extend, funcionarão exatamente com o caso de uso base, como por exemplo o UC05 estendido pelo UC03, porém com outros passos. Já os casos de uso interligados via include, UC07, poderá ser executado para partir do UC08.

#### 4.2.2. Matriz de rastreabilidades

O Quadro 1, desenvolvido nesta seção, representa a matriz de rastreabilidades entre os requisitos funcionais elencados na seção 4.1.3, e o diagrama de atividades presente na seção 4.2.1, deste capítulo.

Quadro 1 – Matriz de rastreabilidades UC x RF

Casos de Uso	Requisitos Funcionais	Descrição
UC01	RF005	Campo para inserção do nome do responsável pela coleta.
UC02	RF009	Possibilidade de escolher os sensores relacionados aos dados que serão monitorados.
UC03	RF012	Pré-apresentação de como os dados serão salvos no banco de dados.
UC04	RF011	Gerenciamento de resultados.
UC05	RF002	Salvar dados obtidos nas coletas
UC06	RF008	Possibilidade de descartar o resultado obtido após uma coleta de dados.
UC07	RF007	Excluir um resultado salvo.
UC08	RF010	Apresenta resultados salvos.
UC09	RF004	Procurar resultados salvos.

**Detalhamento UC02:** Possibilidade de escolher os sensores relacionados aos dados que serão monitorados.

Nome: Umidade, temperatura, pressão, frequência e pH.

Pré-Requisito: Marcar checkbox dos sensores desejados.

Fluxo de eventos:

- Acessar o sistema.
- Inserir o nome do usuário.
- Marcar os sensores.
- Executar os testes.

Ator: Usuário.

#### 4.3. DIAGRAMA DE ATIVIDADES

Atividades, segundo o site Sinônimo, revisto em 2018, significa funcionamento, operação, atuação e execução.

No contexto da UML, o Diagrama de Atividades é um diagrama comportamental, ou seja, que especifica o comportamento do software, e através dele podemos modelar partes do comportamento de um software.

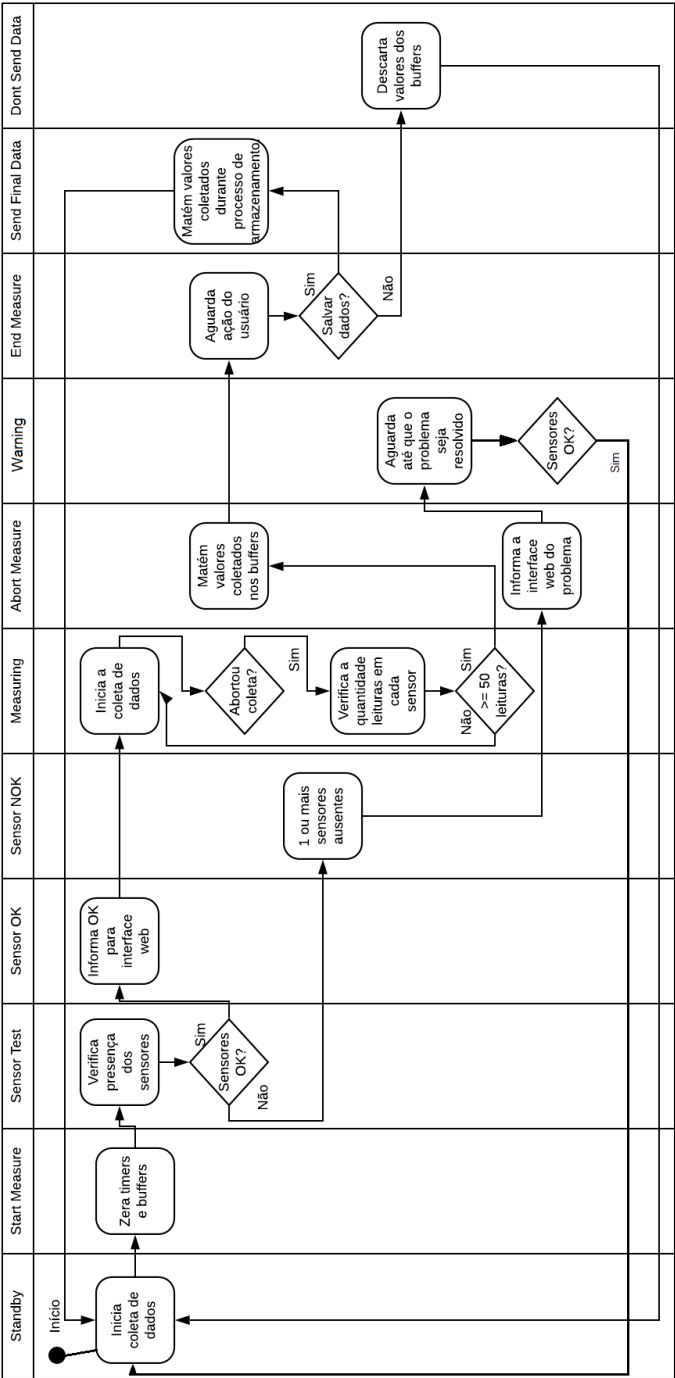
O diagrama de atividades ilustra graficamente como será o funcionamento do software, em nível micro ou macro, como será a execução de alguma de suas partes, como será a atuação do sistema na realidade de negócio na qual ele está inserido.

Em projetos de software utilizamos modelos para representar tanto a estrutura quanto o comportamento do sistema e com bases neles contruir, programar o modelo executável, que é o sistema materializado (VENTURA, 2016).

### 4.3.1. Diagrama de atividades do sistema embarcado

A Figura 8, representa graficamente as atividades do sistema embarcado, com foco nas funções acionadas através da intervenção do usuário na interface web.

Figura 8 – Diagrama de atividades do sistema embarcado



De acordo com a Figura 8, cada raia representa um estado da máquina de estados finitos do software embarcado, sendo que dentro de cada estado será executada apenas suas respectivas atividades.

#### 4.4. MODELAGEM UML

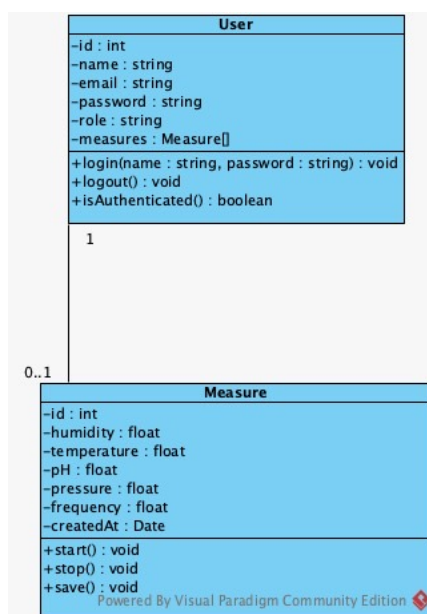
As modelagens do projeto e da proposta de solução utilizarão a Linguagem de Modelagem Unificada (do inglês Unified Modeling Language - UML).

Segundo Fowler (2007), a UML é trata-se de uma família composta por notações gráficas, que ajuda na descrição de projeto de sistemas de software, em particular, aqueles construídos utilizando orientação a objetos.

Larman (2007), diz que a literatura de UML, por ser mais visual, explora a capacidade cerebral de rapidamente abarcar símbolos, unidades e relacionamentos, facilitando a análise dos elementos e relação entre os mesmos dentro de um software.

A Figura 9, apresenta o diagrama UML do back-end desenvolvido para aplicação web.

Figura 9 – Diagrama UML do back-end

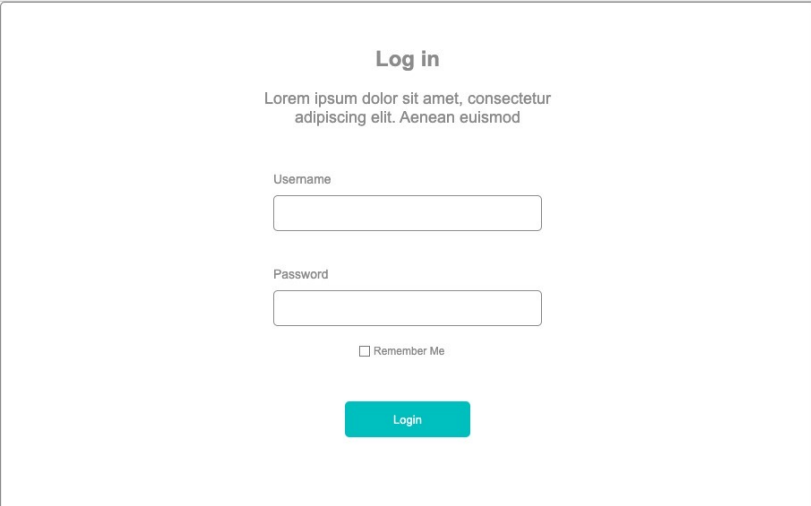


De acordo com a Figura 9, o diagrama UML apresenta os atributos relacionados ao usuário na tabela User, que realizará o teste, bem como os valores obtidos em cada teste representados pelos atributos da tabela Measure.

#### 4.5. A APLICAÇÃO WEB

Esta seção apresenta os protótipos de tela da interface gráfica, bem como as funcionalidades disponibilizadas para os experimentos.

Figura 10 – Protótipo de tela para acessar o sistema



Log in

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod

Username

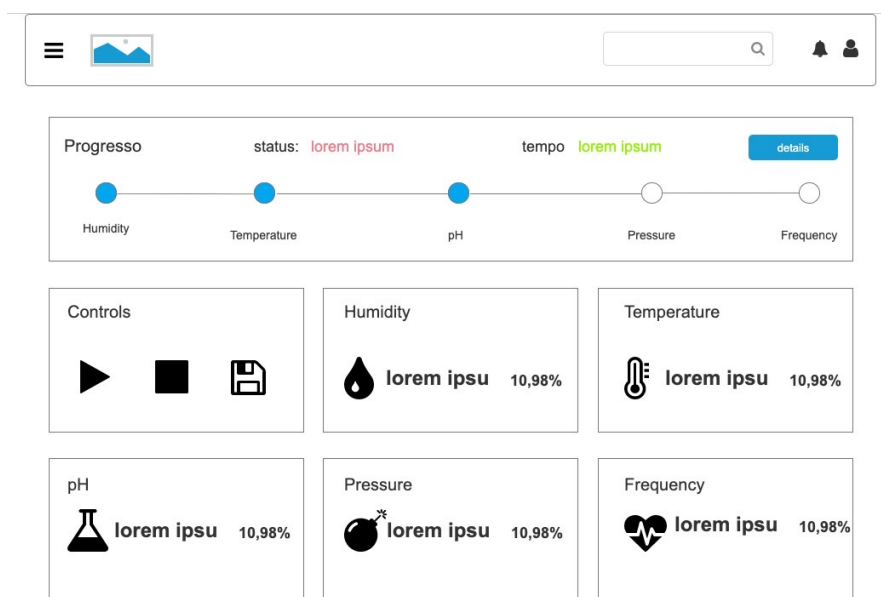
Password

☐ Remember Me

Login

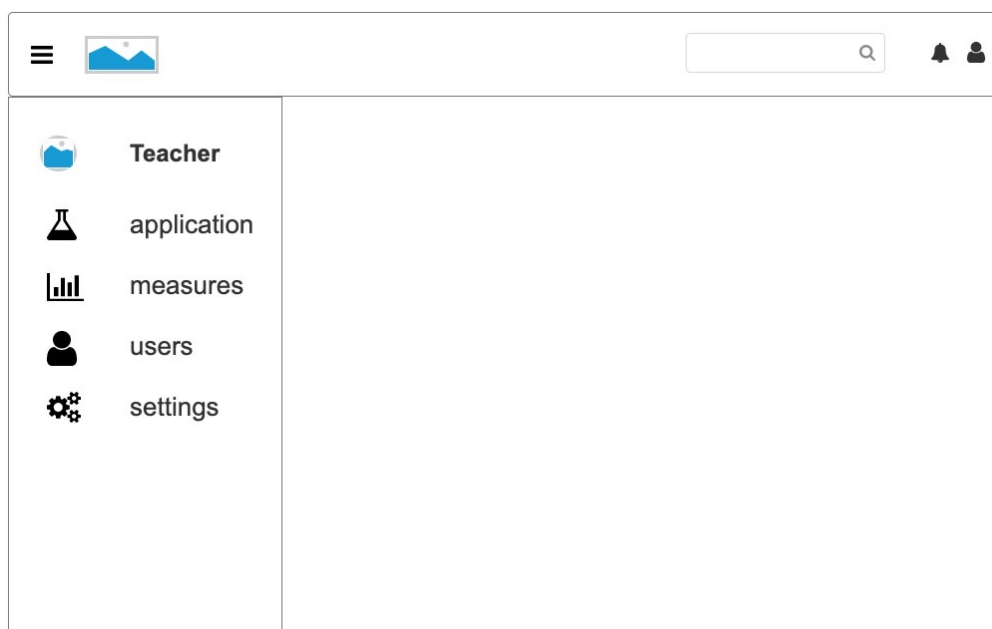
Fonte: os autores, 2020.

Figura 11 – Protótipo de tela para apresentar dados em tempo real



Fonte: os autores, 2020.

Figura 12 – Protótipo de tela para área administrativa e exportação de relatórios



Fonte: os autores, 2020.

A Figuras 10, 11 e 12, representam os protótipos de tela utilizados como base para o desenvolvimento da aplicação web presente no projeto.

## 4.6. O SOFTWARE EMBARCADO

Escrito em linguagem C, o software embarcado deste projeto foi desenvolvido utilizando o modelo computacional de Máquinas de Estados Finitos, do tipo transdutor, ou seja, possui entradas e saídas de dados. A escolha deste modelo, caracteriza-se pela possibilidade de criar estados independentes para cada ação do usuário através da interface gráfica. Além ter o controle sobre os estados, em tempo real, e de modo simultâneo, será realizada a coleta dos dados obtidos em cada sensor e informado ao software embarcado, no respectivo estado.

### 4.6.1. Estados do Sistemas

Neste tópico, são apresentados todos os estados do sistema, bem como as ações do usuário e decisões do próprio sistema permitidas em cada estado. Cada estado, será representado no código escrito em C por um enum, iniciado em 0, totalizando 11 estados.

Quadro 2 - Estados do sistema

ID	Estados	Ações / Decisões
0	STANDBY	Estado que representa o sistema ligado.  Ações: <ul style="list-style-type: none"><li>• Iniciar o teste;</li><li>• Buscar o resultado de um teste no banco de dados;</li></ul> Decisões: <ul style="list-style-type: none"><li>• Trocar para o estado START_MEASURE;</li></ul>
1	START_MEASURE	Estado de inicialização do experimento.  Decisões: <ul style="list-style-type: none"><li>• Zerar os buffers que armazenarão os valores de cada medida por sensor;</li><li>• Zerar timers do escalonador;</li><li>• Trocar para o estado SENSOR_TEST;</li></ul>



2	SENSOR_TEST	<p>Estado que verificar que os sensores estão presentes ou conectados na plataforma.</p> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Se as flags referente a conexão de cada sensor retornar 1, o estado é trocado para SENSOR_OK, senão para SENSOR_NOK;</li> </ul>
3	SENSOR_OK	<p>Estado que assume a presença de todos os sensores.</p> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Troca para o estado MEASURING;</li> <li>• Informa para interface gráfica que as medidas serão realizadas;</li> </ul>
4	SENSOR_NOK	<p>Estado que assume a ausência de um ou mais sensores, e não permite que o experimento seja inicializado.</p> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Troca para o estado ABORT_MEASURE;</li> <li>• Informa a interface gráfica qual(is) sensor(es) está(ão) ausente(s).</li> </ul>
5	ABORT_MEASURE	<p>Estado que aborta o experimento.</p> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Trocar para o estado WARNING se o estado anterior for SENSOR_NOK;</li> <li>• Trocar para o estado END_MEASURE se o estado anterior for MEASURING;</li> </ul>
6	MEASURING	<p>Estado onde são armazenados e apresentados os valores obtidos pelos sensores durante o experimento.</p> <p>Ações:</p> <ul style="list-style-type: none"> <li>• Abortar o teste;</li> </ul> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Armazenar dados obtidos de cada sensor e armazenar em seu respectivo buffer;</li> <li>• Informar para a interface o valor obtido em cada sensor e apresentá-lo em tempo real;</li> <li>• Verificar se a quantidade de vezes lida por cada sensor é maior ou igual a 50 e trocar para o estado ABORT_MEASURE, caso o usuário resolva abortar o teste;</li> </ul>

7	END_MEASURE	<p>Estado onde os sensores param de medir, e último valor de cada sensor permanece apresentado na interface gráfica.</p> <p>Ações:</p> <ul style="list-style-type: none"> <li>• Enviar valores para o banco de dados;</li> <li>• Não enviar valores para o banco de dados;</li> </ul> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Trocar para estado SEND_FINAL_DATA ou DONT_SEND_FINAL_DATA de acordo com a ação do usuário;</li> </ul>
8	SEND_FINAL_DATA	<p>Estado que chama a função responsável por armazenar os valores no banco de dados.</p> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Enviar dados para armazenamento;</li> <li>• Zerar os buffers que armazenarão os valores de cada medida por sensor;</li> <li>• Zerar timers do escalonador;</li> <li>• Trocar para estado STANDBY;</li> </ul>
9	DONT_SEND_DATA	<p>Estado onde a função responsável por armazenar os valores no banco de dados, não é chamada.</p> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Zerar os buffers que armazenarão os valores de cada medida por sensor;</li> <li>• Zerar timers do escalonador;</li> <li>• Trocar para estado STANDBY;</li> </ul>
10	WARNING	<p>Estado que informa a interface gráfica que há um problema com um ou mais sensores.</p> <p>Decisões:</p> <ul style="list-style-type: none"> <li>• Em loop, verifica a flag de conexão de cada sensor;</li> <li>• Caso todas as flags tenha como valor o número 1, troca para o estado STANDBY, senão permanece no estado WARNING;</li> </ul>

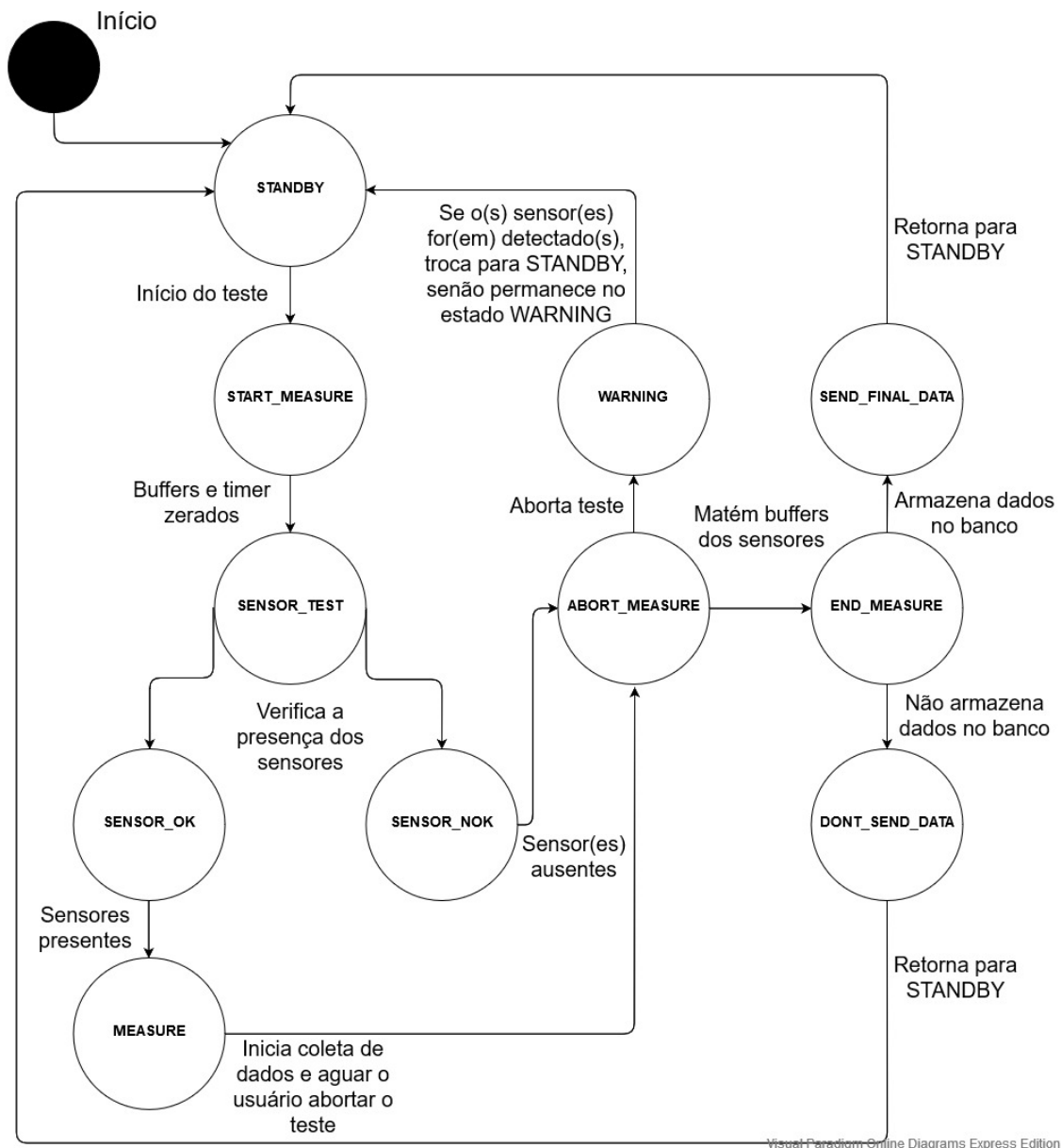
Fonte: os autores, 2020.

De acordo com o Quadro 2, em cada estado estão presentes as possíveis atividades a serem realizadas e comandadas pela própria máquina de estado, com base nos eventos tratados.

#### 4.6.2. Diagrama de estados finitos do sistema

A apresentação do diagrama de estados do sistema, tem como principal objetivo demonstrar a troca de estados de acordo com a ação do usuário e decisões do próprio sistema.

Figura 13 – Diagrama de estados finitos do sistema embarcado



Fonte: dos autores, 2020.

De acordo com a Figura 13, a troca de estados acontece exatamente com as condições pré-definidas em código e com as ações do usuário.

## 4.7. O HARDWARE

Basicamente, hardware, é a parte física de um computador, formado pelos componentes eletrônicos, como por exemplo, circuitos de fios e luz, placas, utensílios, correntes, e qualquer outro material em estado físico, que seja necessário para fazer com que o computador funcione (SIGNIFICADOS, 2017).

O Hardware, é geralmente utilizado por computadores e elementos eletrônicos. Qualquer equipamento físico como, chaves, fechaduras, correntes e peças do próprio computador, são chamados de hardware. O hardware não se limita apenas a computadores pessoais, também está disponível em automóveis, celulares, tablets e etc.

Há vários tipos de hardware, que possuem diferente objetivos e funcionalidades. O hardware de rede, por exemplo, é um equipamento construído com o propósito de possibilitar e gerir equipamentos que estão conectados em rede.

### 4.7.1. O hardware BeagleBone Black

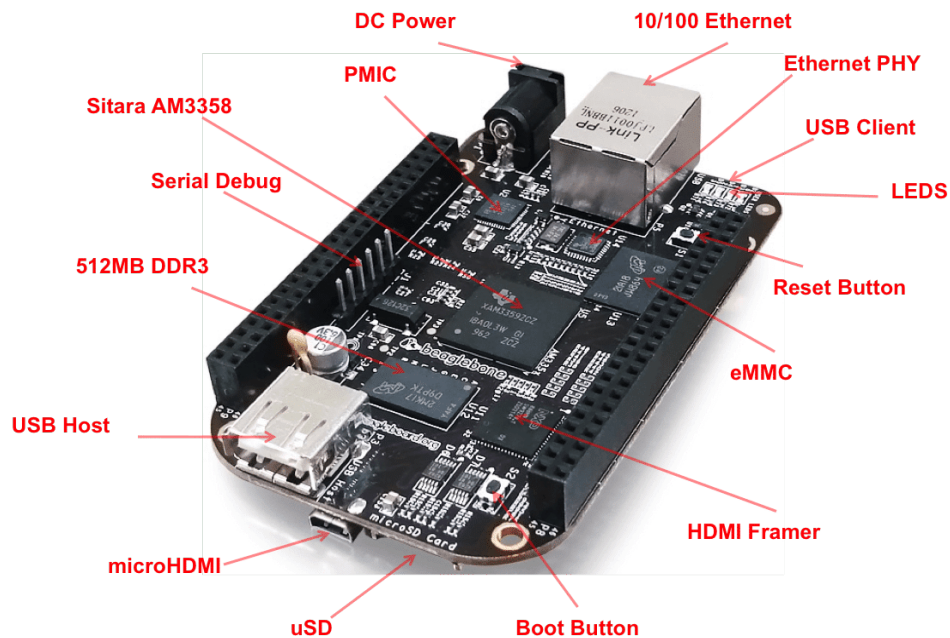
A BeagleBone Black, é um kit de desenvolvimento baseado no processador AM3358 que integra um ARM Cortex-A8 core, rodando a 1GHz e disponibilidade de vários periféricos. Além do processador, a placa possui as interfaces como Ethernet, USB, OTG, cartão TF, serial, JTAG, HDMI, ADC, I2C, SPI, PWM e LCD.

Por se tratar de um hardware robusto, que possibilita o desenvolvimento de variadas aplicações, é compatível com os seguintes sistemas operacionais: Angstrom Linux, Android, Ubuntu, Fedora, Windows Embedded Compact e as distribuições Yocto e Buildroot, para o desenvolvimento de aplicações com Linux embarcado.

#### 4.7.2. Representação física da BeagleBone Black

A Figura 14, apresenta a BeagleBone Black, bem como suas interfaces de comunicação, microprocessador, memória RAM, memória flash e seus periféricos.

Figura 14 - O hardware BeagleBone Black



Fonte: <https://beagleboard.org/black>

- **Interface 10/100 Ethernet:** Porta para conexão com o cabo de rede, para comunicação em rede e internet. Esta interface, é controlada pelo módulo de rede Ethernet PHY, do qual possui o driver para comunicação com o AM3358;
- **Interface USB Client:** Porta utilizada para desenvolvimento e alimentação da BeagleBone através do PC;

- **LEDS:** Os leds soldados na BeagleBone, são utilizados para informar quando o hardware está ligado, bem como, a troca de pacotes utilizando a interface serial;
- **Reset Button:** Botão utilizado apenas para realizar a reinicialização geral do hardware;
- **eMMC:** Este componente eletrônico, trata-se da memória flash da BeagleBone, com capacidade de armazenamento de 2GB, onde encontra-se instalado o sistema operacional e as demais aplicações;
- **HDMI Framer:** Chip responsável por controlar a saída HDMI integrada ao hardware, e que possui o driver para comunicação com o AM3350;
- **Boot Button:** Botão utilizado apenas para inicializar o hardware, quando o mesmo encontra-se desligado, porém, alimentado pela fonte de alimentação;
- **Interface uSD:** Esta interface pode ser utilizada para conectar um cartão de memória MicroSD, com o intuito de aumentar a capacidade de armazenamento do hardware;
- **Interface MicroHDMI:** Esta interface é controlada pelo HDMI Framer, e pode ser utilizada para conectar qualquer periférico de apresentação de imagens, como por exemplo um monitor;
- **Interface USB Host:** Esta interface pode ser utilizada para recepção de dispositivos USB, como por exemplo, um mouse, teclado, pendrive e etc;
- **512MB DDR3:** Este componente eletrônico, trata-se da memória RAM, com 512 megabytes, modelo DDR3, presente no hardware;
- **Interface Serial DEBUG:** Esta interface pode ser utilizada para realizar o monitoramento da troca de pacotes da própria interface serial;

- **Sitara AM3358:** Este componente eletrônico, trata-se do Processador ARM Cortex-A8, de 1GHz presente no hardware;
- **PMIC:** Este componente eletrônico, trata-se do regulador de tensão presente no hardware. Este, é responsável por controlar a alimentação recebida através do plug DC POWER;
- **Plug DC POWER:** Este componente, trata-se da entrada de alimentação do hardware;

Figura 15 - Fonte de alimentação



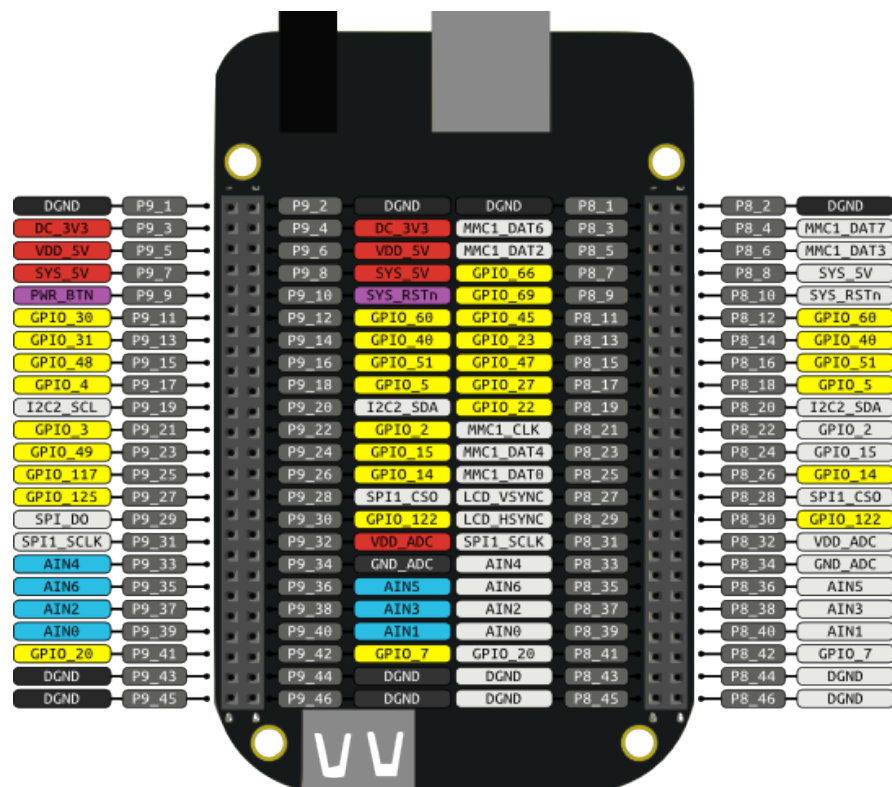
Fonte: os autores, 2020.

- **Fonte de Alimentação:** Fonte de alimentação, figura 15, utilizada para alimentação do hardware. Adaptador AC/DC, entrada 100-240V~50/60Hz 1.0A, saída 5V por 4.0A;

### 4.7.3. Representação gráfica das portas digitais e analógicas da BeagleBone Black

A Figura 16, apresenta os portas/pinos, ou seja, entradas e saídas, também conhecidos no mundo dos sistemas embarcados como GPIO, do inglês General Purpose Input/Output, digitais e analógicas da BeagleBone Black. Estas entradas e saídas, são geralmente utilizadas para realizar acionamentos, bem como coletas de dados, dos periféricos que podem ser conectados na mesma. Os periféricos utilizados neste projeto, são os sensores de Ph, Temperatura, Umidade, Pressão e frequência.

Figura 16 - GPIOs BeagleBone Black



Fonte: <https://beagleboard.org/black>



De acordo com a Figura 16, a BeagleBone Black possui uma vasta quantidade de GPIOs, que possibilita a expansão do projeto com a adição de mais sensores, bem como periféricos se necessário.

#### 4.7.4. pHmetro

O phmetro é um equipamento utilizado para medir o potencial hidrogeniônico ou pH, de uma amostra. O phmetro é construído por um eletrodo e um circuito potenciômetro. A medição ocorre de forma muito simples, ou seja, após o medidor de ph ser calibrado com uma solução tampão, o eletrodo é mergulhado a ponta na amostra, e imediatamente ele produz milivolts que são transformados em escala de pH.

O medidor de pH, figura 17, é um equipamento muito utilizado em laboratório para determinar a concentração de pH em variadas amostras, como por exemplo, queijos, sucos, cervejas, carnes, cremes, água e em soluções gerais.

Figura 17 - pHmetro



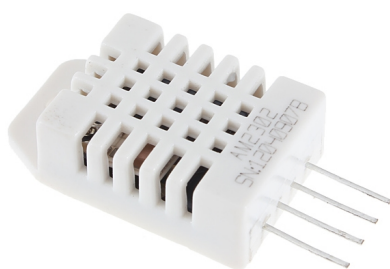
Fonte: os autores, 2020.

Para este projeto, foi utilizado o pHmetro de modelo pH Meter V1.1, fabricado pela DFROBOT. Este modelo necessita de uma tensão de 5 volts para que se tenha o funcionamento esperado, com um range que pode variar de 0 a 14 o nível de pH, com uma temperatura de até 60 graus celsius, com a acurácia de aproximadamente 0.1pH e um tempo de resposta menor ou igual a 1 minuto.

#### **4.7.5. Sensor de umidade e temperatura**

O sensor de umidade e temperatura AM2302-DHT22, figura, 18, permite realizar leituras de temperatura entre -40 a +80 graus celsius e a umidade entre 0 a 100%. Este sensor pode ser utilizado para realizar leituras das respectivas medidas em ambientes abertos e fechados, bem como de um ambiente espectral, no qual realizado neste projeto, em um becker de vidro.

Figura 18 - Sensor de umidade e temperatura



Fonte: os autores, 2020.

A tensão de operação deste sensor pode variar entre 3 a 5 volts com um tempo de resposta de até 2 segundos.

#### 4.7.6. Freqüencímetro

O sensor de pulso monitor, ou frequência, figura 19, CJMCU, opera com uma tensão que pode variar entre 3 a 5 volts, possui um sensor óptico integrado, e com um tempo de resposta de até 2 segundos.

Figura 19 – Freqüencímetro



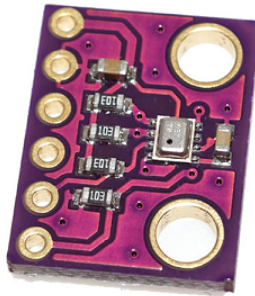
Fonte: os autores, 2020.

Este sensor pode ser utilizado para capturar frequência cardíaca, bem como a frequência espectral em ambientes fechados, caso este, utilizado neste projeto.

#### 4.7.7. Barômetro

O sensor de pressão BMP280, figura 20, opera com uma tensão de 3.3 volts, com um range que pode variar entre 300 a 1100hPa, em temperaturas entre -40 a + 85 graus celsius. Este sensor é geralmente utilizado para medir a pressão atmosférica, temperatura e altitude aproximada do local onde encontra-se instalado.

Figura 20 – Barômetro



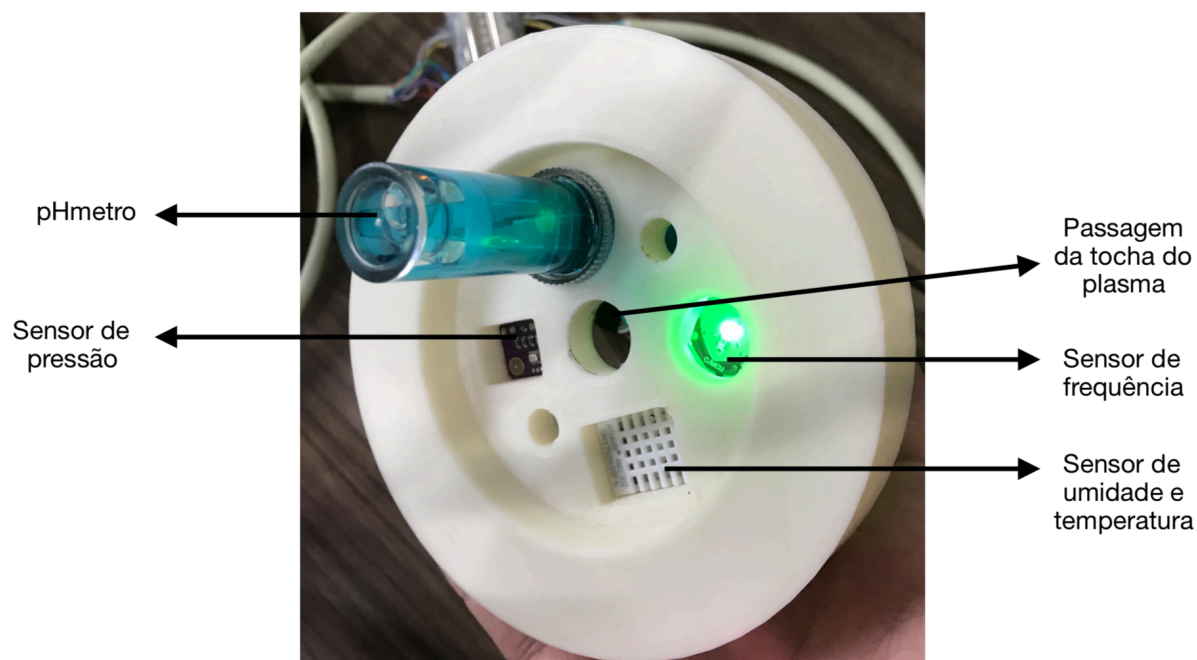
Fonte: os autores, 2020.

Neste projeto, foi utilizado apenas para medir a pressão espectral dentro do becker.

#### **4.7.8. Acoplamento dos sensores ao becker**

Para realizar o acoplamento dos sensores, de forma que fosse possível deixá-los fixos no becker durante os experimentos, foi desenvolvido e impresso em 3D uma tampa, com passagens para os sensores e a tocha de plasma. A Figura 21, apresenta o protótipo definitivo.

Figura 21 - Tampa para o becker



Fonte: os autores, 2020.

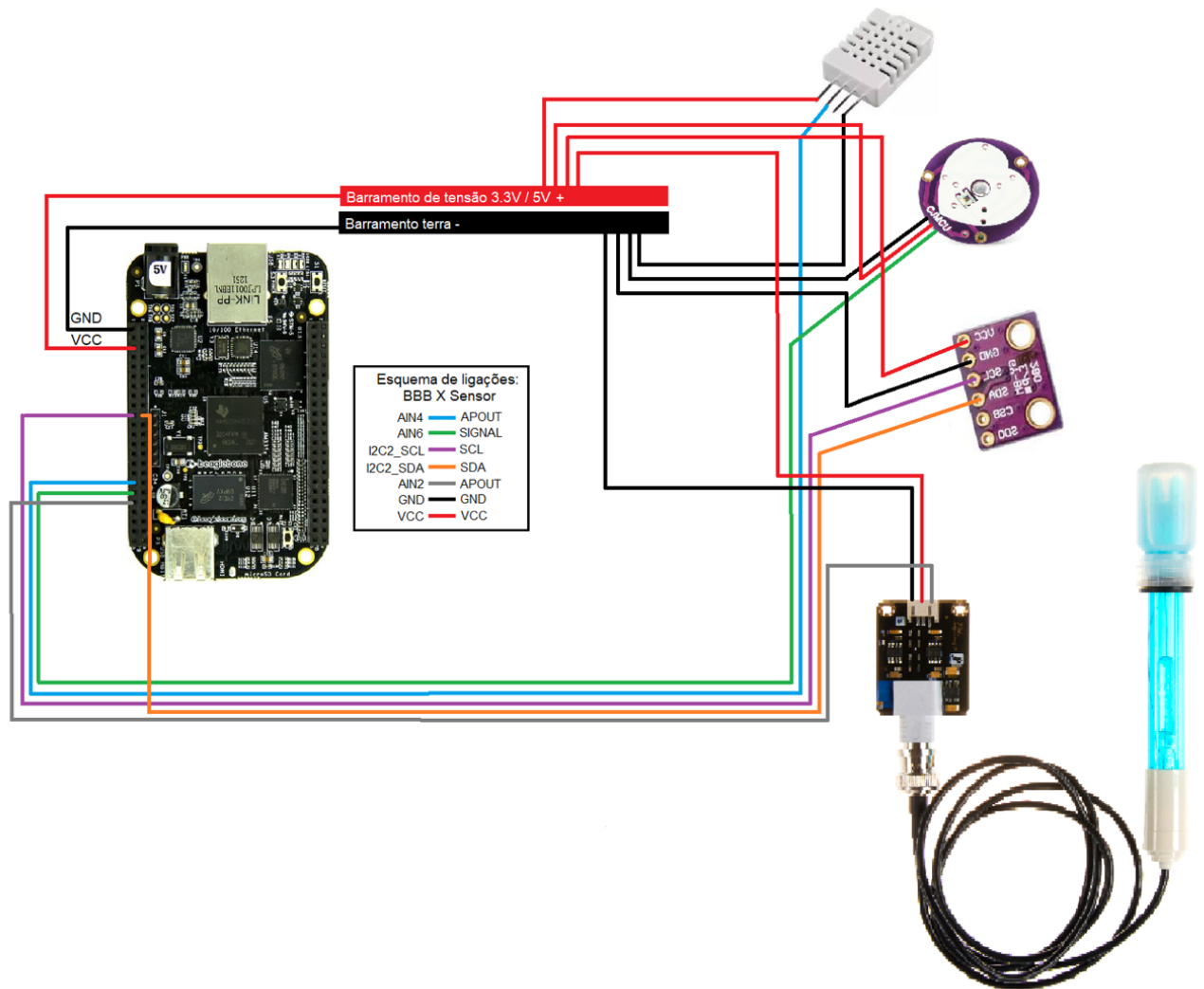
A tampa impressa em 3D possui as seguintes medidas:

- **Circunferência total externa:** 11,0 centímetros de diâmetro;
- **Circunferência total interna:** 7,0 centímetros de diâmetro;
- **Passagem sensor de umidade e temperatura:** 2,0 x 1,5 centímetros;
- **Passagem sensor de pressão:** 1,5 x 1,0 centímetros;
- **Passagem sensor de frequência:** 1,5 centímetros de diâmetro;
- **Passagem pHmetro:** 2,5 centímetros de diâmetro;
- **Passagem da tocha de plasma:** 1,0 centímetro de diâmetro;

#### 4.7.9. Esquema de ligações entre a BeagleBone Black e os Sensores

A figura 22, apresenta o esquema de ligações entre a BeagleBone Black e os sensores, bem como uma legenda demonstrando as portas utilizadas em ambas.

Figura 22 – Esquema de ligações



Fonte: autores, 2020.

O primeiro protótipo desenvolvido para validação do hardware utilizado no projeto, segue exatamente o esquema montando de acordo com a Figura 22.

## 4.8. O BANCO DE DADOS

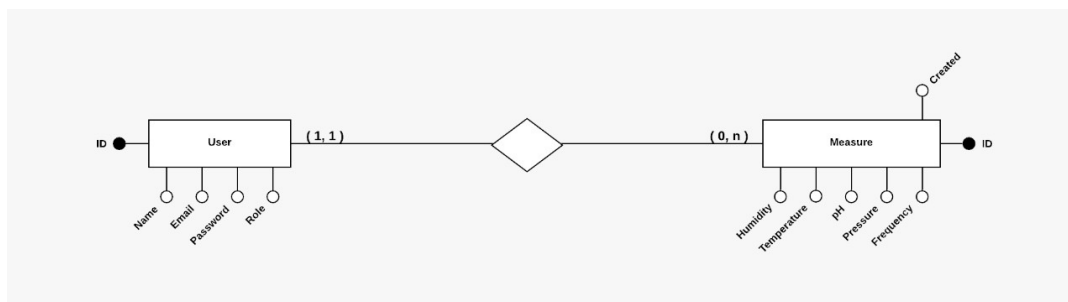
Um banco de dados é uma coleção organizada de informações, ou de dados, estruturados, normalmente armazenados eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas bancos de dados.

Os dados, nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas, em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A grande parte dos bancos de dados usa linguagem de consulta estruturada SQL para escrever e consultar dados (ORACLE, 2020).

### 4.8.1. Modelo conceitual de dados

A modelagem conceitual, baseia-se na análise dos elementos e fenômenos relevantes de uma realidade observada ou imaginada e a posterior formação de um modelo abstrato do corpo de conhecimento adquirido, o modelo entidade-relacionamento ou MER (ALMEIDA, 2018).

Figura 23 – Modelo conceitual de dados do sistema



Fonte: autores, 2020.

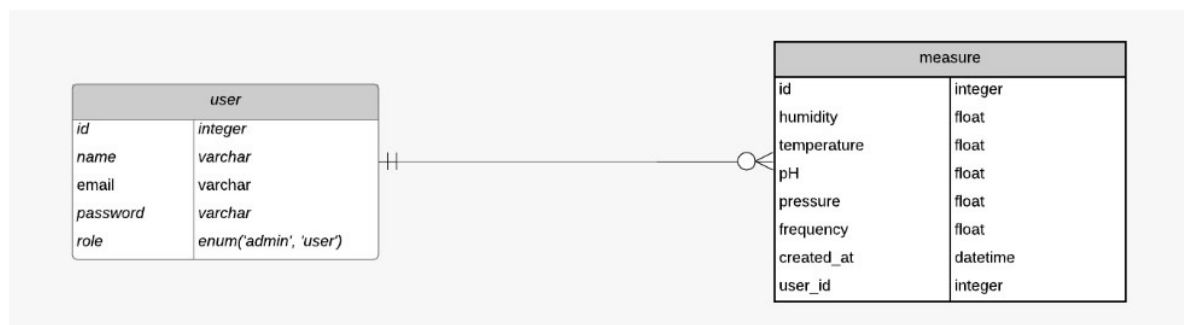
A Figura 23, apresenta o modelo conceitual de dados, implementado com a finalidade de obter os dados durante a utilização do sistema.

#### 4.8.2. Modelo entidade relacionamento

A modelagem entidade relacionamento, ou também chamada por Modelo ER (MER), trata-se de um modelo conceitual utilizado na engenharia de software para descrever os objetos, ou entidades envolvidas em um domínio de negócios, com suas características e como elas se relacionam entre si.

A Figura 24, representa o modelo entidade relacional implementado no sistema.

Figura 24 – Modelo entidade relacional do sistema



Fonte: autores, 2020.

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tablas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais (DEV MEDIA, 2004).



## 5. FERRAMENTAS E RESULTADOS

Neste capítulo são apresentados os resultados obtidos através dos experimentos realizados no laboratório de plasma da Unisul, utilizando o projeto desenvolvido. Também serão apresentadas as ferramentas utilizadas para a construção e testes durante o desenvolvimento do projeto.

### 5.1. DESENVOLVIMENTO

A aplicação Web, escrita em JavaScript e o framework VUE, apresentará os dados em tempo real durante a realização dos experimentos. Através da mesma, será possível iniciar e parar os experimentos, realizar buscas no banco de dados, bem como exportar os resultados obtidos em forma CSV e PDF.

O software embarcado, escrito em linguagem C, é o responsável por coletar os dados dos sensores conectados ao hardware, e enviá-los a aplicação Web, que por sua vez armazena os dados no banco de dados.

O hardware é composto por uma plataforma para desenvolvimento de sistemas embarcados, neste caso a Beaglebone Black, e os sensores para obter temperatura, umidade, PH, frequência e pressão.

No banco de dados em PostgreSQL, serão armazenados os seguintes dados:

- ID do experimento;
- Nome do responsável pelo teste;
- Data e Hora;
- Temperatura;
- Umidade;
- PH;
- Frequência;
- Pressão;

## 5.2. DESENVOLVIMENTO DO SISTEMA EMBARCADO

Nesta seção, é apresentado o modelo de implementação do software embarcado.

A técnica utilizada, referente ao módulo principal do software embarcado foi máquina de estados finitos, representada na Figura 26, do item 5.2.2.

São mostrados também os componentes da máquina de estados, as funções que capturam os dados obtidos através dos sensores, bem como as funções que realizam os cálculos de amostragem.

### 5.2.1. Componentes utilizados na máquina de estados do sistema

A Figura 25, apresenta constantes, variáveis, buffers, enums e funções utilizadas na máquina de estados.

Figura 25 – Componentes da máquina de estados

```
/*CONSTANTES*/
#define TIMES_50_COUNT      50

/*FIM CONSTANTES*/

/*ENUMS*/

typedef enum{
    BY_WEB_MSG              = 0,
    BY_BOARD_MSG            = 1,

    BY_MSG_MAX,
}e_msg_type;

typedef enum{
    ALL_SENSORS_IS_NOT_PRESENT = 0,
    ALL_SENSORS_IS_PRESENT    = 1,

    ALL_SENSORS_MAX,
}e_sensor_presence;

typedef enum{
    MSG_SWT_INIT_TEST        = 0,
    MSG_SWT_ABORT_TEST       = 1,
    MSG_SWT_SAVE_RESULTS     = 2,
    MSG_SWT_DONT_SAVE_RESULTS = 3,

    MSG_SWT_MAX,
}e_msg_switch;

typedef enum{
    STM_MMI_STANDBY          = 0,
    STM_MMI_START_MEASURE    = 1,
    STM_MMI_SENSOR_TEST      = 2,
    STM_MMI_SENSOR_OK        = 3,
    STM_MMI_SENSOR_NOK       = 4,
    STM_MMI_ABORT_MEASURE    = 5,
    STM_MMI_MEASURING        = 6,
    STM_MMI_END_MEASURE      = 7,
    STM_MMI_SEND_FINAL_DATA   = 8,
    STM_MMI_DONT_SEND_DATA    = 9,
    STM_MMI_WARNING           = 10,

    STM_MMI_MAX,
}e_mmi_states;
```

```

/*FIM ENUMS*/

/*VARIÁVIES*/

int mmi_id;
e_mmi_states mmi_st[MMI_NUM_SENSORS];

float getDataPhSensor;
float getTempSensor;
float getDataUmidSensor;
float getDataFrquSensor;
float getDataPressSensor;

float finalBufferDataPhSensor[TIMES_50_COUNT];
float finalBufferDataTempSensor[TIMES_50_COUNT];
float finalBufferDataUmidSensor[TIMES_50_COUNT];
float finalBufferDataFrquSensor[TIMES_50_COUNT];
float finalBufferDataPressSensor[TIMES_50_COUNT];

/*FIM VARIÁVIES*/

/*FUNÇÕES*/

e_sensor_presence p_checkAllFlagSensors(void);
e_msg_switch p_getMsgByWebSock(e_msg_type m_type, char *msg);

void p_memSetAllSensorBuffers(void);
void p_memSetSchedulerTimers(void);
void p_SendMsgToWebSock(e_msg_type m_type, char *msg);

char *p_getStrDataPhSensor(void);
char *p_getStrDataTempSensor(void);
char *p_getStrDataUmidSensor(void);
char *p_getStrDataFreqSensor(void);
char *p_getStrDataPressSensor(void);
char *p_getFinalBufferDataPhSensor(void);
char *p_getFinalBufferDataTempSensor(void);
char *p_getFinalBufferDataUmidSensor(void);
char *p_getFinalBufferDataFreqSensor(void);
char *p_getFinalBufferDataPressSensor(void);

/*FIM FUNÇÕES*/

```

Fonte: autores, 2020.

Os enums representam os tipos de mensagens, status dos sensores e os estados do sistema. As variáveis representam os valores obtidos de cada sensor, os buffers acumulam os valores de cada variável respectiva ao sensor e as funções são chamadas de acordo com o estado.

### 5.2.2. Representação da máquina de estados finitos do sistema desenvolvida em linguagem C

A Figura 26, apresenta a máquina de estados do sistema, realizando a troca de estados, bem como, chamada de funções, utilização de variáveis, troca de mensagens com a aplicação web e a captura de ações do usuário.

Figura 26 - Máquina de estados do sistema

```
void p_mmi_main_stm(void)
{
    switch(mmi_st[mmi_id])
    {
        case STM_MMI_STANDBY:
            last_mmi_st[mmi_id] = mmi_st[mmi_id];

            if(p_getMsgByWebSock(BY_WEB_MSG, NULL) == MSG_SWT_INIT_TEST)
                mmi_st[mmi_id] = STM_MMI_START_MEASURE;
            break;

        case STM_MMI_START_MEASURE:
            p_memSetAllSensorBuffers();
            p_memSetSchedulerTimers();

            last_mmi_st[mmi_id] = mmi_st[mmi_id];
            mmi_st[mmi_id] = STM_MMI_SENSOR_TEST;
            break;

        case STM_MMI_SENSOR_TEST:
            last_mmi_st[mmi_id] = mmi_st[mmi_id];

            if(p_checkAllFlagSensors() == ALL_SENSORS_IS_PRESENT)
                mmi_st[mmi_id] = STM_MMI_SENSOR_OK;
            else if(p_checkAllFlagSensors() != ALL_SENSORS_IS_NOT_PRESENT)
                mmi_st[mmi_id] = STM_MMI_SENSOR_NOK;
            break;

        case STM_MMI_SENSOR_OK:
            p_SendMsgToWebSock(BY_BOARD_MSG, msg);

            last_mmi_st[mmi_id] = mmi_st[mmi_id];
            mmi_st[mmi_id] = STM_MMI_MEASURING;
            break;

        case STM_MMI_SENSOR_NOK:
            p_SendMsgToWebSock(BY_BOARD_MSG, msg);

            last_mmi_st[mmi_id] = mmi_st[mmi_id];
            mmi_st[mmi_id] = STM_MMI_ABORT_MEASURE;
            break;

        case STM_MMI_MEASURING:
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getStrDataPhSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getStrDataTempSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getStrDataUmidSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getStrDataFreqSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getStrDataPressSensor());

            timesCountTest++;
            last_mmi_st[mmi_id] = mmi_st[mmi_id];

            if(p_getMsgByWebSock(BY_WEB_MSG, NULL) == MSG_SWT_ABORT_TEST && timesCountTest >= TIMES_50_COUNT)
            {
                timesCountTest = 0;
                mmi_st[mmi_id] = STM_MMI_ABORT_MEASURE;
            }
            break;

        case STM_MMI_ABORT_MEASURE:
            if(last_mmi_st[mmi_id] == STM_MMI_SENSOR_NOK)
                mmi_st[mmi_id] = STM_MMI_WARNING;
            else if(last_mmi_st[mmi_id] == STM_MMI_MEASURING)
                mmi_st[mmi_id] = STM_MMI_END_MEASURE;

            last_mmi_st[mmi_id] = mmi_st[mmi_id];
            break;

        case STM_MMI_SEND_FINAL_DATA:
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getFinalBufferDataPhSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getFinalBufferDataTempSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getFinalBufferDataUmidSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getFinalBufferDataFreqSensor());
            p_SendMsgToWebSock(BY_BOARD_MSG, p_getFinalBufferDataPressSensor());

            p_memSetAllSensorBuffers();
            p_memSetSchedulerTimers();

            last_mmi_st[mmi_id] = mmi_st[mmi_id];
            mmi_st[mmi_id] = STM_MMI_STANDBY;
            break;

        case STM_MMI_DONT_SEND_DATA:
            p_memSetAllSensorBuffers();
            p_memSetSchedulerTimers();

            last_mmi_st[mmi_id] = mmi_st[mmi_id];
            mmi_st[mmi_id] = STM_MMI_STANDBY;
            break;

        case STM_MMI_END_MEASURE:
            last_mmi_st[mmi_id] = mmi_st[mmi_id];

            if(p_getMsgByWebSock(BY_WEB_MSG, NULL) == MSG_SWT_SAVE_RESULTS)
                mmi_st[mmi_id] = STM_MMI_SEND_FINAL_DATA;
            else if(p_getMsgByWebSock(BY_WEB_MSG, NULL) == MSG_SWT_DONT_SAVE_RESULTS)
                mmi_st[mmi_id] = STM_MMI_DONT_SEND_DATA;

            break;

        case STM_MMI_WARNING:
            if(p_checkAllFlagSensors() == ALL_SENSORS_IS_PRESENT)
            {
                last_mmi_st[mmi_id] = mmi_st[mmi_id];
                mmi_st[mmi_id] = STM_MMI_STANDBY;
            }
            break;

        default:
            /* no data.. */
            break;
    }
}
```

De acordo com a Figura 26, que representa o código escrito em linguagem C da máquina de estados do sistema, é possível observar os estados, e dentro de cada um a troca de mensagens necessárias para o funcionamento da mesma, bem como as funções utilizadas.

### 5.2.3. Funções para captura de dados dos sensores

A Figura 27, apresenta as funções que obtém o valor capturado por cada sensor, quando a máquina de estados está no estado MEASURING. Após capturar o valor, o mesmo é salvo no seu respectivo buffer e transformado no tipo char\* para ser apresentado na interface web.

Figura 27 - Funções para captura de dados

```
/*Função para obter o Ph*/
char *p_getStrDataPhSensor(void)
{
    getDataPhSensor = (analogRead(GPIO_PH_SENSOR) / ((1024.0 * 5.0) * 1.45));
    finalBufferDataPhSensor[timesCountTest] = getDataPhSensor;
    return (char *)getDataPhSensor;
}

/*Função para obter a temperatura*/
char *p_getStrDataTempSensor(void)
{
    getDataTempSensor = dht.readTemperature();
    finalBufferDataTempSensor[timesCountTest] = getDataTempSensor;
    return (char *)getDataTempSensor;
}

/*Função para obter a Umidade*/
char *p_getStrDataHumidSensor(void)
{
    getDataHumidSensor = dht.readHumidity();
    finalBufferDataHumidSensor[timesCountTest] = getDataHumidSensor;
    return (char *)getDataHumidSensor;
}

/*Função para obter a Frequência*/
char *p_getStrDataFreqSensor(void)
{
    getDataFreqSensor = analogRead(GPIO_FREQ_SENSOR);
    finalBufferDataFreqSensor[timesCountTest] = getDataFreqSensor;
    return (char *)getDataFreqSensor;
}

/*Função para obter a Pressão*/
char *p_getStrDataPressSensor(void)
{
    getDataPressSensor = analogRead(GPIO_FREQ_SENSOR);
    finalBufferDataPressSensor[timesCountTest] = getDataPressSensor;
    return (char *)getDataPressSensor;
}
```

De acordo com a Figura 27, a única função que necessita de uma fórmula para apresentar o valor obtido pelo sensor, é a *char \*p\_getStrDataPhSensor(void)*, responsável por obter o valor do pH e retornar para o sistema no tipo *char \**. A fórmula consiste no seguinte cálculo:

- 1024: represente o valor máximo em bytes trafegado pelo GPIO;
- 5.0: a tensão necessária para alimentar o sensor;
- 1.45: a corrente necessária para o funcionamento correto do sensor;

A partir do momento em que o sensor obtém o valor através da função *analogRead(GPIO\_PH\_SENSOR)*, será feito a divisão do mesmo por  $((1024 * 5.0) * 1.45)$  e obtem-se o valor do pH dentro do range esperado.

As demais funções, *char \*p\_getStrDataTempSensor(void)*, *char \*p\_getStrDataHumidSensor(void)*, *char \*p\_getStrDataFreqSensor(void)* e *char \*p\_getStrDataPressSensor(void)*, dispensam a implementação de uma fórmula para obter o valor esperado.

#### **5.2.4. Funções para o cálculo final de amostragem**

A Figura 28, apresenta as funções que realizam a soma do valor acumulado no buffer de cada sensor, durante 50 vezes. Ao concluir a soma, o valor total é dividido por 50, obtendo assim a média final. Este valor é transformado no tipo *char\** para ser apresentado na interface web, e podendo ser salvo no banco de dados.

Figura 28 - Funções para cálculos de amostragem

```
/*Função que calcula a amostragem final da Temperatura*/
char *p_getFinalBufferDataTempSensor(void)
{
    float totalBuffer;

    for(int i = 0; i <= TIMES_50_COUNT; i++)
    {
        totalBuffer += finalBufferDataTempSensor[TIMES_50_COUNT];
    }

    return (char *)totalBuffer/TIMES_50_COUNT;
}

/*Função que calcula a amostragem final da Umidade*/
char *p_getFinalBufferDataHumidSensor(void)
{
    float totalBuffer;

    for(int i = 0; i <= TIMES_50_COUNT; i++)
    {
        totalBuffer += finalBufferDataHumidSensor[TIMES_50_COUNT];
    }

    return (char *)totalBuffer/TIMES_50_COUNT;
}

/*Função que calcula a amostragem final da Frequência*/
char *p_getFinalBufferDataFreqSensor(void);
{
    float totalBuffer;

    for(int i = 0; i <= TIMES_50_COUNT; i++)
    {
        totalBuffer += finalBufferDataFreqSensor[TIMES_50_COUNT];
    }

    return (char *)totalBuffer/TIMES_50_COUNT;
}

/*Função que calcula a amostragem final da Pressão*/
char *p_getFinalBufferDataPressSensor(void)
{
    float totalBuffer;

    for(int i = 0; i <= TIMES_50_COUNT; i++)
    {
        totalBuffer += finalBufferDataPressSensor[TIMES_50_COUNT];
    }

    return (char *)totalBuffer/TIMES_50_COUNT;
}
```

Fonte: autores, 2020.

De acordo com a Figura 28, e conforme mencionado acima, cada buffer irá acumular 50 vezes o valor obtido de cada sensor, totalizando 250 leituras referentes aos 5 sensores.

### 5.3. FERRAMENTAS DE APOIO

Nesta seção, serão apresentadas as ferramentas utilizadas para o desenvolvimento completo do projeto.

#### 5.3.1. JavaScript

O JavaScript é uma linguagem de programação que permite implementar funcionalidades mais complexas em páginas web. Sempre que uma página web faz mais do que apenas mostrar informações estáticas, em tempo real, ela mostra conteúdos atualizados, mapas interativos, animações gráficas, vídeos e etc. Para o front-end e back-end deste projeto, o javascript foi escolhido, por ser uma das linguagens mais utilizadas para o desenvolvimento de aplicações web, pela vasta gama de recursos e pelo desempenho da própria aplicação (MDN, 2020).

Informações obtidas através do site: <https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript>

#### 5.3.2. Node.js

Uma ferramenta Javascript feita a partir do mecanismo Javascríp V8 do navegador Chrome, que faz com que uma aplicação rode em tempo de execução. Pelo fato de utilizar o Javascript, o mesmo acaba sendo eficiente e leve. Além disso, o Node.js é conhecido mundialmente como o maior ecossistemas de bibliotecas código aberto. Esta ferramenta foi selecionada a fim de ser utilizadas alguma bibliotecas no front-end (OPENJS, 2020).

Informações obtidas através do site: <https://nodejs.org/en/>



### 5.3.3. Express

Express é um popular framework web estruturado, escrito em JavaScript que roda sobre o ambiente node.js em tempo de execução. O Express fornece uma camada fina de recursos fundamentais para aplicativos web, sem obscurecer os recursos do Node.js (OPENJS, 2020).

Informações obtidas através do site: <https://expressjs.com/pt-br/>

### 5.3.4. PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional (SGBDOR) baseado no POSTGRES Versão 4.2 desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O POSTGRES foi o pioneiro de vários conceitos que somente se tornaram disponíveis muito mais tarde em alguns sistemas de banco de dados comerciais (SOURCEFORGE, 2009).

Informações obtidas através do site: <https://pgdocptbr.sourceforge.net/pg82/intro-what-is.html>

### 5.3.5. Knex

Knex é um módulo que traz uma série de funções que simplificam operações de dados, evitando o uso direto do SQL. As operações de inserção, consulta, alteração e remoção de dados podem ser usadas a partir de funções que retornam promissas, o que torna o código mais claro e objetivo (OPENJS, 2020).

Informações obtidas através do site: <https://books.google.com.br/primeiros-passos-com-node-js-pg82>

### 5.3.6. Vue.js

Vue.js, é um framework progressivo para a construção de interfaces de usuário. Ao contrário de outros frameworks monolíticos, Vue foi projetado desde sua concepção para ser adotável incrementalmente. A biblioteca principal é focada exclusivamente na camada visual, sendo fácil de adotar e integrar com outras bibliotecas ou projetos existentes. Por outro lado, Vue também é perfeitamente capaz de dar poder a sofisticadas Single-Page-Applications quando usado em conjunto com ferramentas modernas e bibliotecas de apoio (VUEJS, 2020). Informações obtidas através do site: <https://br.vuejs.org/v2/guide/index.html>

### 5.3.7. Vue router

A característica mais marcante de uma Single page Application é a velocidade. Os elementos de interface já estão presentes e quando sua estrutura de servidor é bem montada, a sensação que o usuário tem ao usar a aplicação é a de que ela está instalada no próprio computador, tão rápida que é a resposta.

Um componente essencial para uma Single Page Application é o Router, que é responsável por montar/esconder um ou mais elementos dependendo da URL que se acessa no browser. Esta é a única responsabilidade do router, o que não significa que seja uma simples ferramenta (VUEJS, 2020).

Informações obtidas através do site: <https://vuejs-brasil.com.br/vue-router/>

### 5.3.8. Vuex

O Vuex, é um padrão de gerenciamento de estado e biblioteca para aplicativos Vue.js. Ele server como um store centralizado para todos os componentes em uma aplicação, com regras garantindo que o estado só possa ser mudado de forma previsível. Ele também se integra com a extensão oficial do Vue devtools para fornecer recursos avançados sem configurações adicionais, como depuração viajando pelo histórico de estado e exportação/importação de registros de estado em determinado momento (VUEJS, 2020).

Informações obtidas através do site: <https://vuex.vuejs.org/ptbr/>

### 5.3.9. Vuetify

O Vuetify é um framework responsivo em Vue, baseado Material Design. Possui uma boa gama de componentes e uma documentação sólida. Para utiliza-lo, é basicamente necessário ter instalado o Node, versão 8 ou superior, e de um editor de textos como o Visual Studio Code por exemplo. Tendo esses itens em funcionamento, projetos Vuetify podem ser criados escolhendo os mais diversos templates existentes (VUEJS, 2020).

Informações obtidas através do site: <https://vuejs-brasil.com.br/conheca-o-vuetify-tutorial-dicas-parte-1/>

### 5.3.10. Axios

Axios é um cliente HTTP, que funciona tanto no browser como no Node.js. A biblioteca é basicamente uma API que sabe interagir tanto com XMLHttpRequest quanto com a interface http do node. Isso significa que o mesmo código utilizado para fazer requisições Ajax no browser também funciona no servidor. Além disso, as requisições feitas através da biblioteca

retornam uma premissa, compatível com a nova versão do Javascript - ES6 (BERNARDES, 2015).

Informações obtidas através do site: <http://codeheaven.io/how-to-use-axios-as-your-http-client-pt/>

### **5.3.11. VSCODE**

VSCODE é um editor de código-fonte desenvolvido pela Microsoft, para Windows, Linux e MacOS. Ele inclui suporte para depuração, controle git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código. Ele também é customizável, fazendo que os usuários possam mudar o tema do editor, teclas de atalho e preferências. Ele é um software livre e de código aberto, apesar do download oficial estar sob uma licença proprietária (WIKIPEDIA, 2020).

Informações obtidas através do site: [https://pt.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://pt.wikipedia.org/wiki/Visual_Studio_Code)

### **5.3.12. Postman**

É uma ferramenta que tem como objeto testar os serviços RESTfull, por meio do envio de requisições HTTP e da análise do seu retorno. Com ele é possível consumir facilmente serviços locais e na internet, enviando dados e efetuando testes sobre as respostas das requisições. O uso de Web APIs, vem crescendo e o Postman nos testes desse tipo de projeto, bem como permite a desenvolvedores analisarem o funcionamento dos serviços externos, a fim de saber como consumi-los (DEV MEDIA, 2019).

Informações obtidas através do site: <https://www.devmedia.com.br/testando-apis-web-com-o-postman/37264>

### **5.3.13. Chrome**

É um dos mais populares navegadores disponíveis para Windows, Linux, MacOS, Android e iOS, mas não possui versão para Windows Phone. Como seu nome deixa claro, o browser é desenvolvido pelo Google, é grátis e tem versão em português. A primeira versão foi liberada em 2 de setembro de 2008 e recebe atualizações constantemente (TECHTUDO, 2016). Informações obtidas através do site: <https://www.techtudo.com.br/tudo-sobre/google-chrome.html>

### **5.3.14. Chrome devtools**

É um conjunto de ferramentas de autoria e depuração de web incorporado ao Google Chrome. O devtools pode ser usado para iterar, debugar e criar o perfil de sua aplicação web (GOOGLE, 2020).

Informações obtidas através do site: <https://developers.google.com/web/tools/chrome-devtools?hl=pt-br>

### **5.3.15. Vue devtools**

É uma ferramenta desenvolvida para facilitar o processo de desenvolvimento de aplicações que utilizam o Vue.js, funciona como uma extensão do browser e serve para visualizar o estado de cada componente que está sendo renderizado (PICOLLO, 2020).

Informações obtidas através do site: <https://blog.geekhunter.com.br/vue-js-so-vejo-vantagens-e-voce/>

### **5.3.16. Eclipse**

É uma ferramenta IDE que compreende vários tipos de linguagem e que aceita a instalação de plugins para emular o desenvolvimento da plataforma. Uma das principais vantagens é o uso do SWT, e a forte orientação ao desenvolvimento baseado em plugins ampliando o suporte do desenvolvedor com centenas deles que procuram atender as diferentes necessidades. Para este projeto, foi utilizado a linguagem C (DEV MEDIA, 2012).

Informações obtidas através do site: <https://www.devmedia.com.br/conhecendo-o-eclipse-uma-apresentacao-detalhada-da-ide/25589>

### **5.3.17. Linguagem C**

A linguagem C foi criada com um propósito: ser utilizada no desenvolvimento de uma nova versão do sistema operacional Unix. A primeira versão Unix utilizava Assembly. Então pode-se dizer que desde o princípio, C foi uma linguagem criada por programadores para programadores.

A linguagem C é considerada de propósito geral, ou seja, é uma linguagem capaz de usada para praticamente qualquer tipo de projeto. É extremamente portátil, ou seja, um programa em linguagem C pode ser facilmente usado em qualquer plataforma. Utilizando o C, podemos criar sistemas operacionais, aplicativos de todos os tipos, drivers e outros controladores de dispositivos, programar microcontroladores e etc (CASAVELLA, 2018).

Informações obtidas através do site: <http://linguagemc.com.br/o-que-e-linguagem-c/>

#### 5.4. RESULTADOS OBTIDOS UTILIZANDO O SISTEMA

Esta seção apresenta os resultados obtidos utilizando o sistema, sem a interface web e banco de dados, apenas apresentando os dados em tempo real.

Para realização dos testes no Laboratório de Plasma da Unisul, bem como validar o funcionamento do sistema desenvolvido, utilizamos um display gráfico de tamanho 20 x 2 conectado a BeagleBone Black apresentando os dados em tempo real.

Os resultados apresentados abaixo, foram obtidos com um de 15 minutos de coleta de dados, em um Becker com 30 mililitros de água contaminada com os efluentes da extração de carvão.

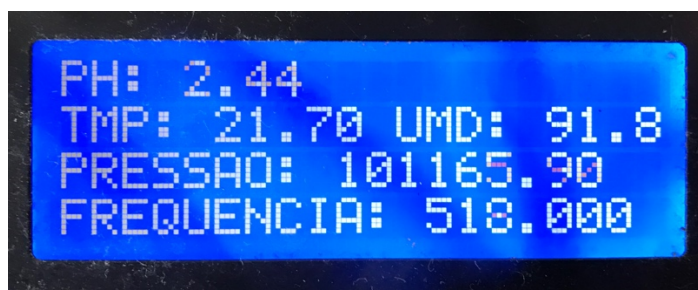
Quadro 3 – Início da coleta

Parâmetros iniciais				
<b>Tempo de teste:</b> 0' 50'' (cinquenta segundos)				
pH	Temperatura	Umidade	Pressão	Frequência
2.44	21.70	91.80	101165.90	518.000

Fonte: autores, 2020.

O Quadro 3, bem como a Figura 29, representam o início da coleta de dados, em um tempo de teste de 50 segundos, de um total de 15 minutos, tempo total do teste.

Figura 29 – Início da coleta



Fonte: autores, 2020.

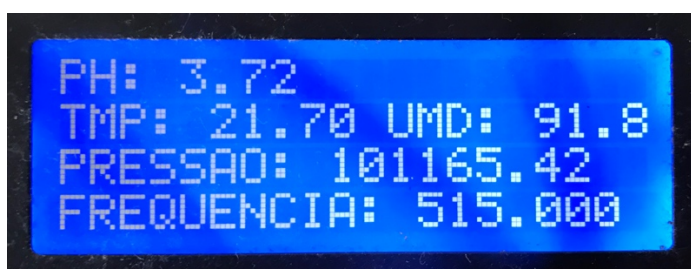
Quadro 4 – Coleta intermediária I

Parâmetros intermediários				
<b>Tempo de teste:</b> 3' 77" (três minutos e setenta e sete segundos)				
pH	Temperatura	Umidade	Pressão	Frequência
3.72	21.70	91.80	101165.42	515.000

Fonte: autores, 2020.

O Quadro 4, bem como a Figura 30, apresentam os resultados obtidos após 3 minutos e 77 segundos de coleta, em tempo real. É possível salientar a diferença no valor do pH.

Figura 30 – Coleta intermediária I



Fonte: autores, 2020.

Quadro 5 – Coleta intermediária II

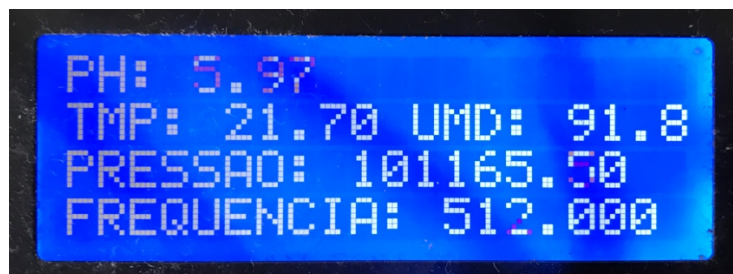
Parâmetros intermediários				
<b>Tempo de teste:</b> 11' 25" (onze minutos e vinte e cinco segundos)				
pH	Temperatura	Umidade	Pressão	Frequência
5.97	21.70	91.80	101165.50	512.000

Fonte: autores, 2020.

O Quadro 5, bem como a Figura 31, representam os resultados obtidos após 11 minutos e 25 segundos de coleta, também em tempo real.



Figura 31 – Coleta intermediária II



Fonte: autores, 2020.

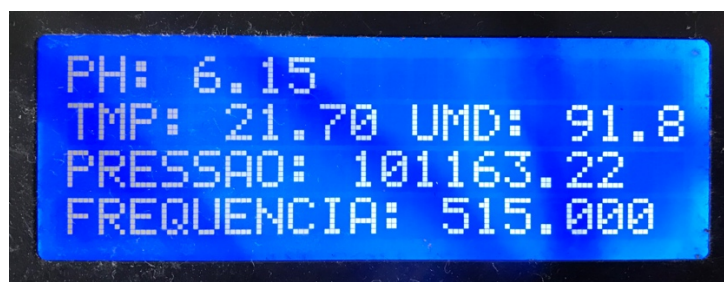
Quadro 6 – Fim da coleta

Parâmetros finais				
<b>Tempo de teste:</b> 15' 11'' (quinze minutos e onze segundos)				
pH	Temperatura	Umidade	Pressão	Frequência
6.15	21.70	91.80	101163.22	515.000

Fonte: autores, 2020.

O Quadro 6, bem como a Figura 32, representam os resultados obtidos após 15 minutos e 11 segundos de coleta, também em tempo real e fim do teste. De acordo com os Quadros e Figuras apresentados acima, houve mudanças expressivas no pH, por se tratar da remediação causada pelo plasma na água em análise.

Figura 32 – Fim da coleta



Fonte: autores, 2020.

Com base nos resultados apresentados, e avaliados pelo laboratório, observa-se que o sistema se mostrou eficaz para obtenção dos dados necessários durante o processo de remediação dos efluentes causados pela extração de carvão mineral.

Este teste foi repetido outras vezes com mais amostras de água e os resultados foram semelhantes, alternando a quantidade de água e tempo de coleta.

## 5.5. APLICAÇÃO WEB

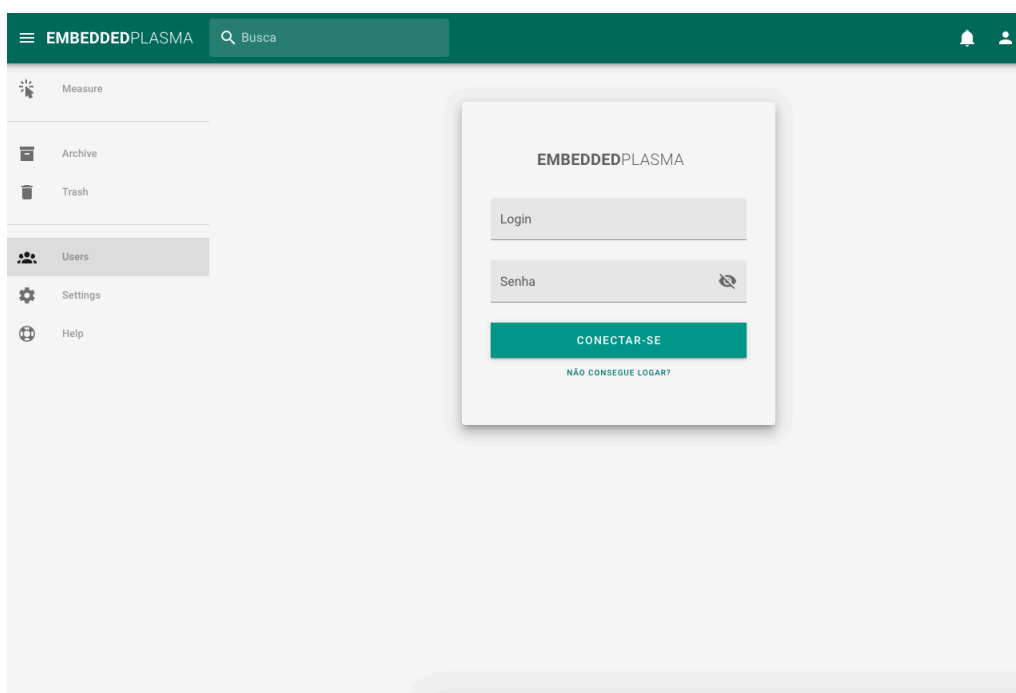
Esta seção apresenta a aplicação web desenvolvida para compor o projeto, juntamente com o sistema embarcado e o banco de dados.

De acordo com os requisitos funcionais abordados no item 4.1.3, a aplicação foi desenvolvida seguindo o diagrama de casos de uso apresentado no item 4.2.1, figura 7, pertencentes ao capítulo 4.

As imagens a seguir, foram capturadas da aplicação web desenvolvida, implementada na BeagleBone Black e em comunicação com os demais componentes do sistema.

A figura 33, representa o acesso principal ao sistema, solicitando ao usuário a inserção de suas credenciais de acesso.

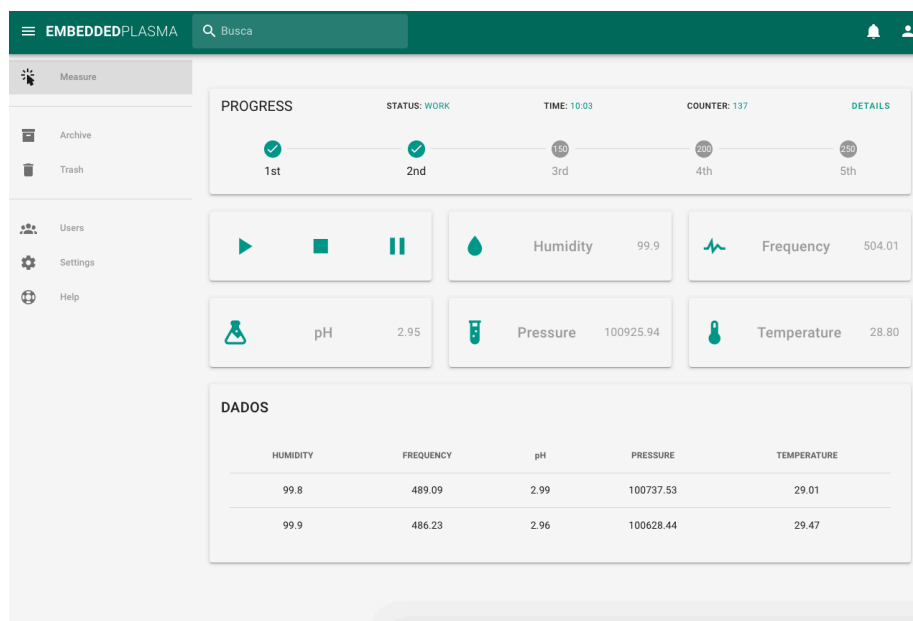
Figura 33 – Tela de login



Fonte: autores, 2020.

Após o acesso, a Figura 34 representa a página principal do sistema, onde o usuário poderá acessar qualquer opção do menu a sua esquerda, bem como inicializar a coleta de dados através do botão play, na cor verde.

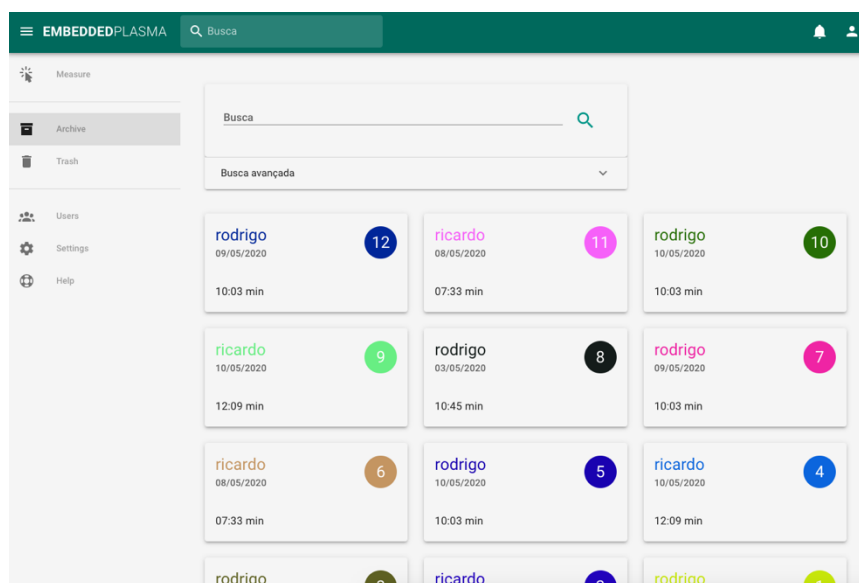
Figura 34 – Tela de coleta de dados



Fonte: autores, 2020.

Na Figura 35, é apresentado o menu Archive, onde ficam registradas todas as coletas realizadas.

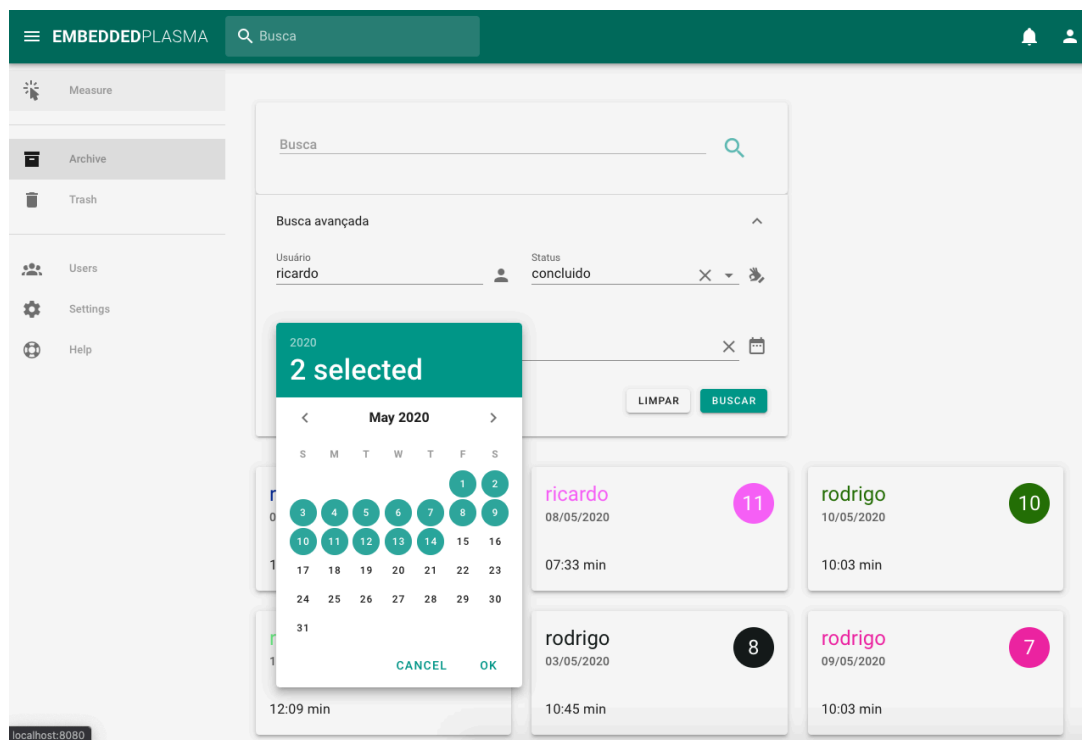
Figura 35 – Tela de registro de coletas



Fonte: autores, 2020.

Ainda no menu Archive, a Figura 36 representa opções para busca de registro de coleta de dados, através de data, usuário e status.

Figura 36 – Tela de busca de coletas de dados



Fonte: autores, 2020.

Ao obter os resultados procurados, além de poder exportá-los do sistema, o usuário também poderá removê-los se achar necessário.

## 5.6. APLICACAÇÃO DAS FERRAMENTAS DE APOIO

Esta seção apresenta alguns trechos de códigos, arquivos de configurações e testes desenvolvidos para a construção da aplicação web, utilizando algumas das ferramentas de apoio abordadas no item 5.3 deste capítulo.

Figura 37 – JavaScript / Cadastro de usuários

```
1 const bcrypt = require('bcrypt');
2
3 module.exports = app => {
4   const { existOrError, notExistsOrError, equalsOrError } = app.api.validators;
5
6   const encryptPassword = password => {
7     const salt = bcrypt.genSaltSync(10);
8     return bcrypt.hashSync(password, salt);
9   };
10
11   const add = async (req, res) => {
12     const user = { ...req.body }
13
14     try {
15       // You: 3 days ago - add user to backend
16       existOrError(user.name, 'Nome não informado. ');
17       existOrError(user.email, 'Email não informado. ');
18       existOrError(user.password, 'Senha não informada. ');
19       existOrError(user.confirmPassword, 'Confirmação de senha inválida. ');
20       equalsOrError(user.password, user.confirmPassword, 'Senhas não conferem. ');
21
22       const userFromDb = await app.db('users')
23         .where({ email: user.email }).first();
24
25       notExistsOrError(userFromDb, 'Usuário já cadastrado. ');
26     } catch (msg) {
27       return res.status(422).send(msg);
28     }
29
30     user.password = encryptPassword(user.password);
31     delete user.confirmPassword;
32
33     await app.db('users')
34       .insert(user)
35       .then(_ => res.status(201).send())
36       .catch(error => res.status(500).send(error));
37   };
38
39   const edit = async (req, res) => {
40     const user = { ...req.body }
41     user.id = req.params.id
42
43     try {
44       existOrError(user.name, 'Nome não informado. ');
45       existOrError(user.email, 'Email não informado. ');
46       existOrError(user.password, 'Senha não informada. ');
47       existOrError(user.confirmPassword, 'Confirmação de senha inválida. ');
48       equalsOrError(user.password, user.confirmPassword, 'Senhas não conferem. ');
49     } catch (msg) {
50       return res.status(422).send(msg);
51     }
52
53     user.password = encryptPassword(user.password);
```

Fonte: autores, 2020.

A Figura 37, apresenta um código escrito em JavaScript que representa o SQLBuilder da aplicação, ou seja, é o responsável por construir as queries principais para o gerenciamento de usuários no banco de dados.

Figura 38 – Node.js / Arquivo de configuração de dependências

```
1 {
2   "name": "backend",
3   "version": "0.1.0",
4   "description": "backend do tcc",
5   "main": "index.js",
6   "scripts": {
7     "start": "nodemon",
8     "production": "pm2 start index.js --name knowledge-backend"
9   },
10  "author": "Rodrigo Mendes Vieira",
11  "license": "ISC",
12  "devDependencies": {
13    "nodemon": "^2.0.3"
14  },
15  "dependencies": {
16    "bcrypt": "^4.0.1",
17    "body-parser": "^1.19.0",
18    "consign": "^0.1.6",
19    "cors": "^2.8.5",
20    "express": "^4.17.1",
21    "jwt-simple": "^0.5.6",
22    "knex": "^0.21.1",
23    "moment": "^2.25.3",
24    "passport": "^0.4.1",
25    "passport-jwt": "^4.0.0",
26    "pg": "^8.1.0"
27  }
28 }
```

Fonte: autores, 2020.

A Figura 38, representa o arquivo de configurações de dependências do Node.js, ou seja, a inclusão das bibliotecas presentes na aplicação bem como as bibliotecas e aplicativos utilizados no desenvolvimento da aplicação web.

Figura 39 – Express

```
1 const admin = require('./admin');
2
3 module.exports = app => {
4   app.post('/signup', app.api.user.add);
5   app.post('/signin', app.api.auth.signin);
6   app.post('/validateToken', app.api.auth.validateToken);
7
8   app.route('/users')
9     .all(app.config.passport.authenticate())
10    .post(admin(app.api.user.any))
11    .get(admin(app.api.user.get));
12
13   app.route('/users/:id')
14     .all(app.config.passport.authenticate())
15    .get(admin(app.api.user.getById))
16    .put(admin(app.api.user.edit))
17    .delete(admin(app.api.user.remove));
18 }
19
```

Fonte: autores, 2020.

A Figura 39, representa o arquivo de configuração de rotas do webservice, referente aos usuários cadastros na aplicação.

Figura 40 – PostgreSQL / Coleta de dados

🔍 Search for item...

Items

Queries

History

▼ Functions

▼ Tables

📄 data

📄 knex\_migrations

📄 knex\_migrations\_lock

📄 measure

📄 users

Name	data	Primary	id	Column name... (16f)			
#	column_name	data_type	is_nullable	check	column_default	foreign_key	comment
1	id	int4	NO	⌵ NULL	nextval('data_id_seq::regclass')	⌵ EMPTY	→ NULL
2	humidity	float4	NO	⌵ NULL	NULL	⌵ EMPTY	→ NULL
3	temperature	float4	NO	⌵ NULL	NULL	⌵ EMPTY	→ NULL
4	pH	float4	NO	⌵ NULL	NULL	⌵ EMPTY	→ NULL
5	pressure	float4	NO	⌵ NULL	NULL	⌵ EMPTY	→ NULL
6	frequency	float4	NO	⌵ NULL	NULL	⌵ EMPTY	→ NULL
7	measure_id	int4	NO	⌵ NULL	NULL	⌵ public.measure(id)	→ NULL

Fonte: autores, 2020.

A Figura 40, apresenta a estrutura da tabela measure presente no banco de dados da aplicação. Esta tabela receberá dos valores obtidos pelos sensores.

Figura 41 – Knex / Querys para dados de usuário

```

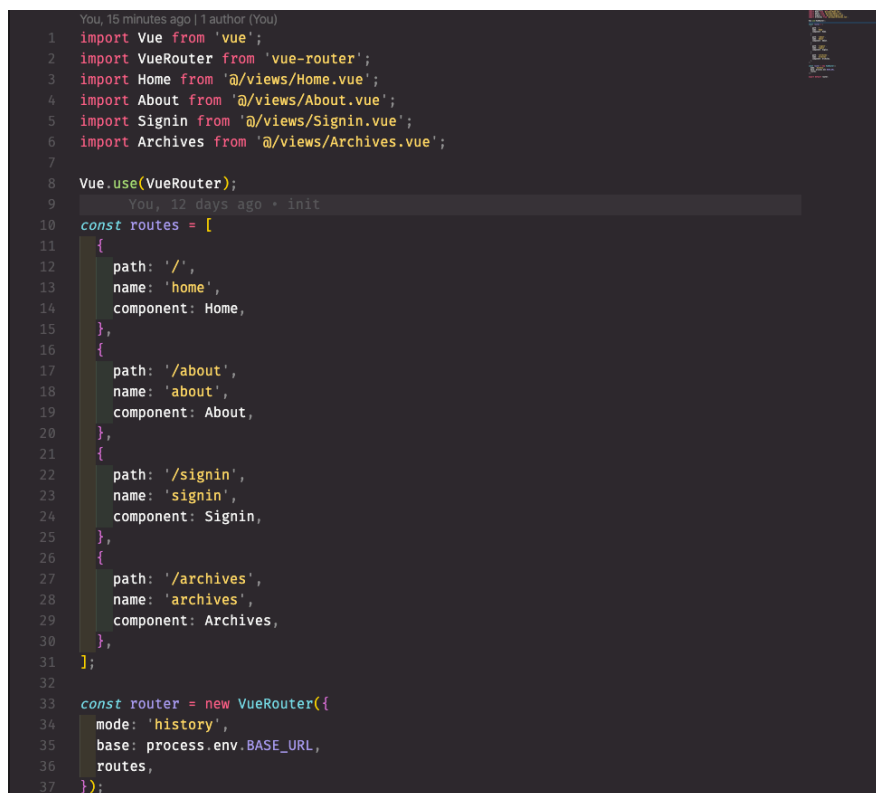
5 module.exports = app => {
6   const signin = async (req, res) => {
7     if (!req.body.email || !req.body.password) {
8       return res.status(400).send('Informe usuário e senha!');
9     }
10
11     const user = await app.db('users')
12       .where({ email: req.body.email })
13       .first();
14     if (!user) return res.status(400).send('Usuário não encontrado');
15
16     const isMatch = bcrypt.compareSync(req.body.password, user.password);
17     if (!isMatch) return res.status(400).send('Email/Senha inválidos!');
18
19     const now = Math.Floor(Date.now() / 1000);
20
21     const payload = {
22       id: user.id,
23       name: user.name,
24       email: user.email,
25       admin: user.admin,
26       iat: now,
27       exp: now + (60 * 60 * 24)
28     };
29
30     res.json({
31       ...payload,
32       token: jwt.encode(payload, authSecret)
33     });
34   };
35
36   const validateToken = async (req, res) => {
37     const userData = req.body || null;
38
39     try {
40       if (userData) {
41         const token = jwt.decode(userData.token, authSecret);
42
43         if (new Date(token.exp * 1000) > new Date()) {
44           return res.send(true);
45         }
46       } catch (e) {
47         // problema com token(possível erro de authSecret)
48       }
49
50       res.send(false);
51     }
52
53     return { signin, validateToken };
54   }
55 }

```

Fonte: autores, 2020

A Figura 41, representa o processo de autenticação e validação do usuário cadastrado ao realizar o acesso a aplicação.

Figura 42 – Vue Router / Rotas da interface



```
1 import Vue from 'vue';
2 import VueRouter from 'vue-router';
3 import Home from '@views/Home.vue';
4 import About from '@views/About.vue';
5 import Signin from '@views/Signin.vue';
6 import Archives from '@views/Archives.vue';
7
8 Vue.use(VueRouter);
9
10 const routes = [
11   {
12     path: '/',
13     name: 'home',
14     component: Home,
15   },
16   {
17     path: '/about',
18     name: 'about',
19     component: About,
20   },
21   {
22     path: '/signin',
23     name: 'signin',
24     component: Signin,
25   },
26   {
27     path: '/archives',
28     name: 'archives',
29     component: Archives,
30   },
31 ];
32
33 const router = new VueRouter({
34   mode: 'history',
35   base: process.env.BASE_URL,
36   routes,
37 });
```

Fonte: autores, 2020

A Figura 42, representa o arquivo de configurações de rotas de toda aplicação, escrito em JavaScript.



Figura 43 – Vuetify / Botões Play, Pause e Stop

```
You, 19 minutes ago | 1 author (You)
1 <template>
2   <v-card class="grey lighten-4 d-flex justify-space-between">
3     <v-tooltip top>
4       <template v-slot:activator="{ on }">
5         <v-btn v-on="on" class="ma-5" color="teal" tile fab icon>
6           <v-icon x-large>mdi-play</v-icon>
7         </v-btn>
8       </template>
9       <span>Play</span>
10    </v-tooltip>
11
12    <v-tooltip top>
13      <template v-slot:activator="{ on }">
14        <v-btn v-on="on" class="ma-5" color="teal" tile fab icon>
15          <v-icon x-large>mdi-stop</v-icon>
16        </v-btn>
17      </template>
18      <span>Stop</span>
19    </v-tooltip>
20
21    <v-tooltip top>
22      <template v-slot:activator="{ on }">
23        <v-btn v-on="on" class="ma-5" color="teal" tile fab icon>
24          <v-icon x-large>mdi-pause</v-icon>
25        </v-btn>
26      </template>
27      <span>Pause</span>
28    </v-tooltip>
29  </v-card>
30 </template>
31
32 <script>
33 export default {
34   name: 'CardCommands',
35 };
36 </script>
37
```

Fonte: autores, 2020.

A Figura 43, representa o componente vue.js utilizado para iniciar, parar e cancelar o processo de coleta de dados, podendo ser visualizado na Figura 33.

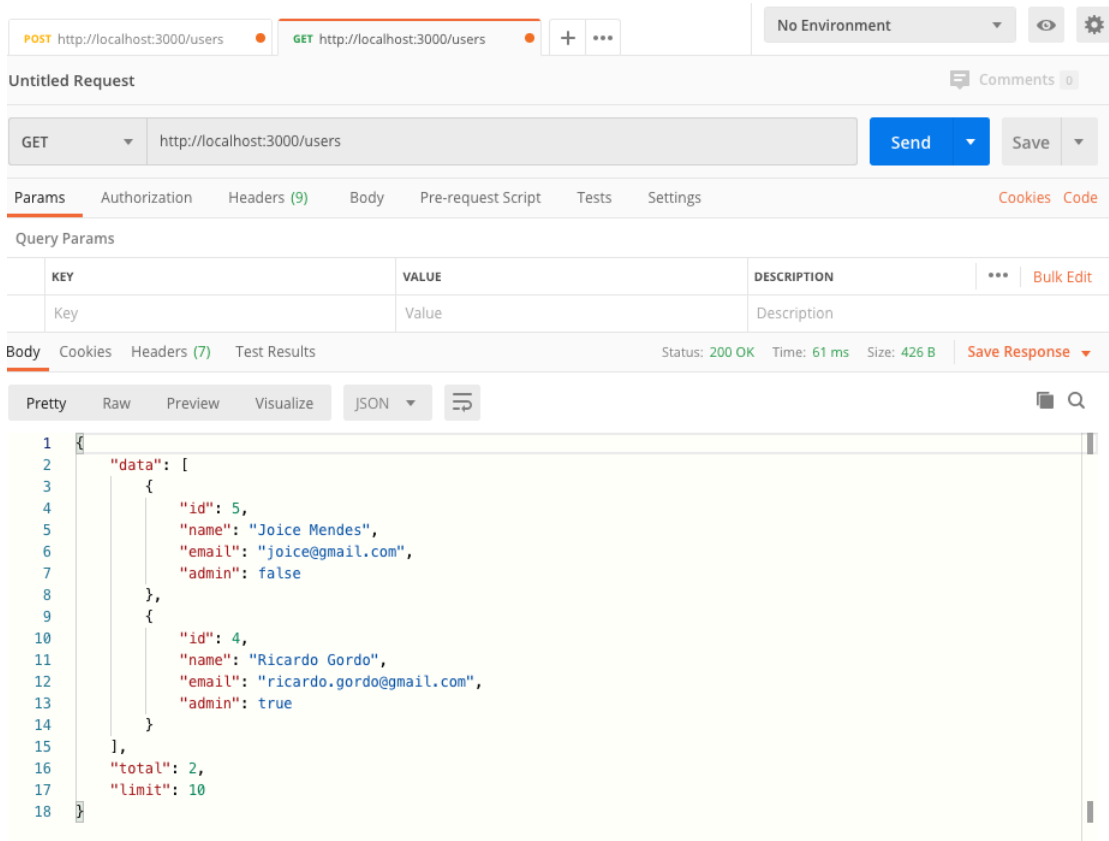
Figura 44 – Axios

```
Too, a few seconds ago | 3 authors (Rodrigo Mendes Vieira and others)
1 import Vue from "vue";
2 import axios from "axios";
3 import SessionStorage from '@services/session-storage';
4
5 axios.interceptors.request.use((config) => {
6   if (!config.url.endsWith('token')) {
7     config.headers.common['ws_token'] = SessionStorage.get('token');
8   }
9
10  return config;
11 }, (error) => {
12  return Promise.reject(error);
13 });
14
15 Vue.use({
16   install(Vue) {
17     Vue.prototype.$http = axios;
18   }
19 });
20
21 export default() => {
22   return axios.create({
23     baseURL: 'http://localhost:8080/api',
24     withCredentials: false,
25     headers: {
26       'Accept': 'application/json',
27       'Content-Type': 'application/json'
28     }
29   })
30 }
31
```

Fonte: autores, 2020.

A Figura 44, representa o módulo que realiza a comunicação entre o front-end e o back-end da aplicação. Ou seja, é responsável por capturar as informações no webservice e entregar ao front-end.

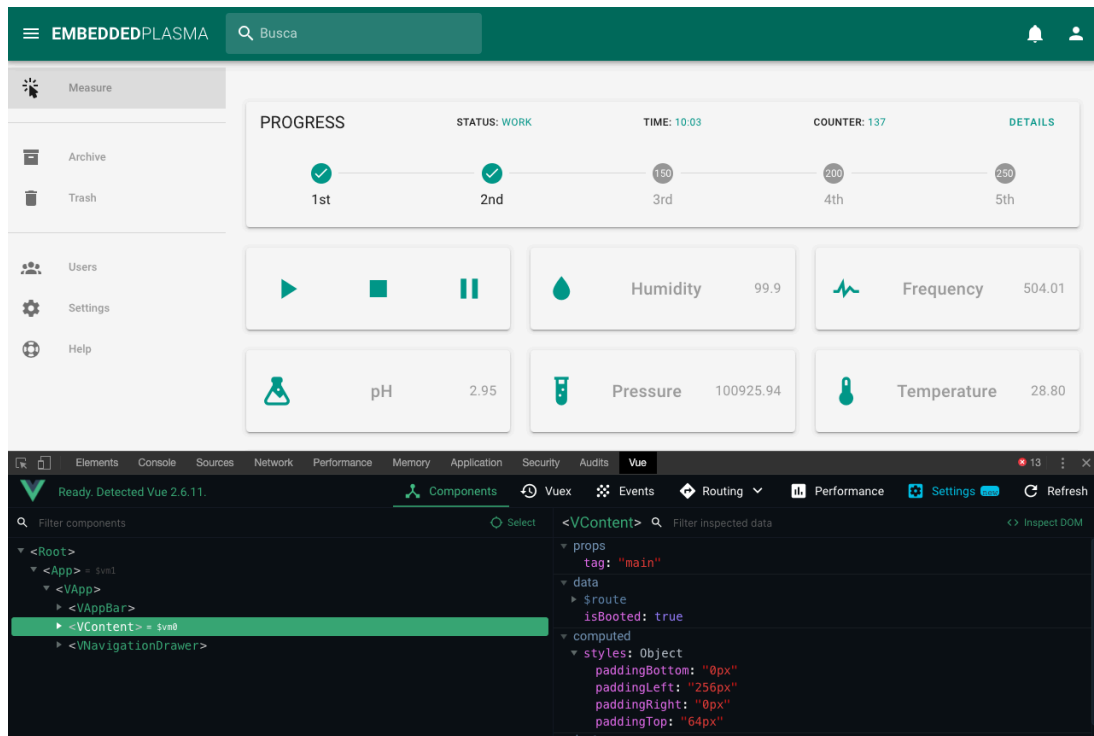
Figura 45 – Postman



Fonte: autores, 2020.

A Figura 45, representa a utilização do Postman para realizar o teste dos endpoints do webservice.

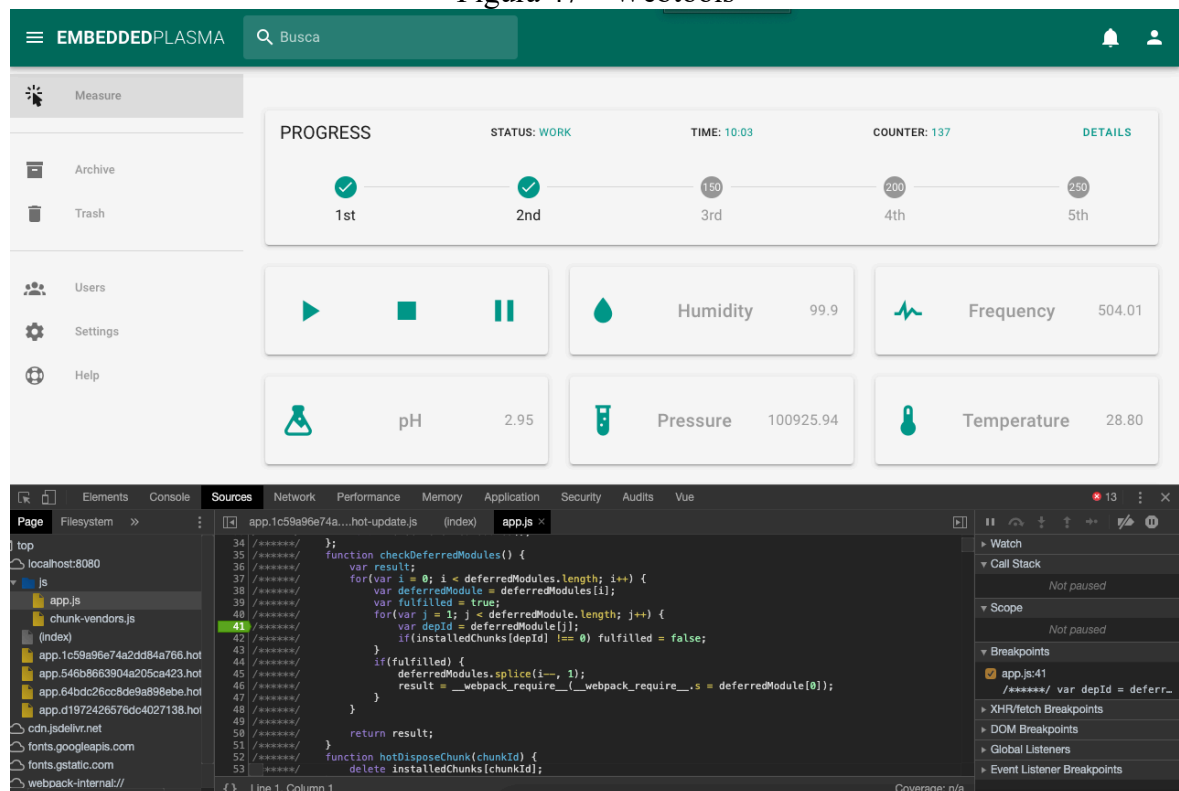
Figura 46 – Vuetools



Fonte: autores, 2020.

A Figura 46, representa a ferramenta utilizada para realizar os testes relacionados ao front-end da aplicação, com foco no vue.js.

Figura 47 – Webtools



Fonte: autores, 2020.

A Figura 47, representa a ferramenta utilizada para realizar os testes relacionados ao back-end e front-end da aplicação de forma geral.

## 5.7. AVALIAÇÃO DO PROJETO

Com a finalidade de avaliar o processo de desenvolvimento do projeto, a realização dos testes feitos em laboratório e os resultados obtidos, esta seção apresenta um questionário respondido pela Professora Dra. Anelise Leal Vieira Cubas, responsável pelo Laboratório de Plasma da Unisul, orientadora da pesquisa que impulsionou o desenvolvimento deste projeto bem como coorientadora do presente trabalho.

### **1 - Quais foram os benefícios de obter-se os dados em tempo real durante a coleta?**

R: O plasma é um processo muito instável, qualquer alteração de vazão de gás ou até mesmo de corrente ou tensão elétrica, pode influenciar nos parâmetros medidos, daí a importância de fazer uma medição em tempo real, pois consegue detectar os parâmetros no momento que estão acontecendo e não de tempos em tempos como é realizado de forma manual.

### **2 - A possibilidade de obter-se dados diferentes em simultâneo contribuíram para o desempenho dos testes?**

R: Sim com certeza, do contrário tínhamos que ficar alternando equipamentos para conseguir medir todos os parâmetros e a instabilidade do plasma proporciona resultados diferentes em tempos diferentes de medida, eles sendo feitos ao mesmo tempo da um resultados mais exato.

### **3 - A possibilidade de realizar a coleta de dados sem a necessidade de parar o processo de remediação dos efluentes da água foi eficaz?**

R: Trabalhamos na maior parte das vezes com análises microbiológicas (tratamento de água, desinfecção de alimentos...) que envolvem microorganismos e a manipulação para medição manual de parâmetros pode ser uma forma de contaminação das amostras, sendo assim a medição remota em tempo real facilita muito evitando possibilidade de contaminação das amostras.

**4 - O sistema embarcado, bem como o protótipo da solução adequou-se com facilidade aos equipamentos do laboratório utilizados para a coleta de dados? Exemplo: becker, retaor..**

R: Sim, na verdade fizemos uma tampa da impressora 3 D com baixo custo e os sensores foram colocados nessa tampa que se adaptou bem a reator de plasma.

**5 - Os dados obtidos pelos sensores utilizados para a construção do protótipo foram coerentes com o esperado?**

R: Sim, os dados são os esperados para a atmosfera de plasma.

**6 - Comparando os dados obtidos utilizando outros equipamentos do laboratório, e os dados obtidos com o protótipo desenvolvido, há semelhanças?**

R: Na verdade esses dados ainda não foram aferidos para comparar com os equipamentos que trabalhamos.

**7 - O protótipo desenvolvido demonstrou confiança para o uso contínuo no laboratório?**

R: O ideal era fazer os testes para aferir os sensores que não conseguimos fazer ainda, mas pelo que tudo indica os sensores são bem confiáveis.

**8 - A possibilidade de armazenamento das coletas de dados para futuras consultas irá contribuir para trabalhos realizados no laboratório?**

R: Muito, isso é fundamental, pois na pesquisa o que mais prezamos é o armazenamento de dados que são trabalhados para melhoramento do processo.

**9 - A opção de operar o protótipo desenvolvido através de uma interface gráfica, irá contribuir para os trabalhos realizados no laboratório?**

R: Sim, facilita depois para trabalhar os dados podendo fazer tabelas e gráficos com maior rapidez que se fossem medidos de forma manual.

**10 - Há melhorias observadas que podem ser implementadas no futuro ao protótipo/pesquisa desenvolvidos?**

R: Acredito que colocar mais alguns sensores e aferir os sensores.

**11 - De forma geral, qual sua avaliação sobre o desenvolvimento deste projeto?**

R: Achei o projeto excelente, e pode ajudar muito nas pesquisas com reatores de plasma, a medição dos parâmetros de forma remota e em tempo real é um grande avanço para as nossas pesquisas com o reator de plasma.

**12 - Como foi a experiência no desenvolvimento deste projeto em conjuntos com os alunos do curso de sistema de informação e o laboratório de plasma?**

R: Foi muito boa, pois agregou os conhecimentos em sistemas embarcados nos nossos reatores de plasma que foram fundamentais para o melhoramento do sistema de controle do processo a plasma.

**13 – Este projeto visa o desenvolvimento de novos projetos capazes de colaborar com os trabalhos realizados no laboratório?**

R: Sim com certeza, agora pretendemos implantar o sistema em todos os nossos reatores de plasma.

## 5.8. CUSTO PARA O DESENVOLVIMENTO DO HARDWARE

Para realizar o desenvolvimento do protótipo de hardware, foi necessário um investimento feito pelo laboratório de plasma.

O quadro 7, apresenta todos os itens adquiridos, bem como os valores unitários e total.

Quadro 7 – Investimento do protótipo

Quantidade	Produto	Valor R\$
01	BeagleBone Black	R\$ 406, 93
01	Fonte de Alimentação 5V/4.0A	R\$ 60,00
01	pHmetro	R\$ 108,00
01	Sensor de umidade e temperatura	R\$ 55,00
01	Frequencímetro	R\$ 26,00
01	Barômetro	R\$ 13,90

01	Impressão da tampara para o becker	R\$ 10,00
30	Conectores diversos	R\$ 30,00
20 metros	Fios	R\$ 26,00
<b>TOTAL</b>		<b>R\$ 735,83</b>

O pHmetro, foi adquirido através de um site de vendas de equipamentos eletrônicos de fora do país. A tampa do Becker, fornecida por um aluno parceiro do laboratório, e os demais equipamentos, foram adquiridos de uma loja virtual, também de equipamentos eletrônicos, situada na Grande Florianópolis.

## 6. CONCLUSÃO

Neste capítulo, são abordados as conclusões e trabalhos futuros, relacionados ao projeto proposto inicialmente. Ainda, os resultados obtidos, desde a problemática inicial, até a avaliação do protótipo pela professora responsável pelo laboratório de plasma. Também são apresentadas novas funcionalidades para o sistema, sendo consideradas como trabalhos futuros.

### 6.1. CONCLUSÕES

De acordo com os objetivos específicos, a proposta de solução baseada em software e hardware, na qual envolve os sensores necessários para o monitoramento do processo de tratamento da água de mina, foi validada com base nos valores obtidos pelos sensores e pelas fórmulas implementadas no software embarcado, comprovando a eficácia do protótipo funcional validado no laboratório.

O projeto desenvolvido com a finalidade de automatizar processos realizados manualmente, previne alterações nos experimentos realizados com o reator, contribuindo com os trabalhos realizados no laboratório.

Trata-se de um novo modelo de ferramenta desenvolvida com a união de setores da tecnologia da informação, eletrônica e engenharia química, a fim de contribuir com as mais variadas áreas e suas problemáticas.

Tendo em vista a problemática principal, levantada pelo laboratório de plasma da Unisul e que levou o desenvolvimento deste projeto, ou seja, realizar a coleta dos dados em simultâneo ao processo de remediações de água de mina com o reator de plasma, sem a necessidade de parar o sistema e sem correr o risco de contaminar a água em tratamento com outras possíveis bactérias, tornando a solução desenvolvida eficaz a sua principal proposta de solução.

De acordo com a avaliação realizada pela Professora Anelise, responsável pelo laboratório, e respondendo a pergunta presente na problemática deste trabalho, sendo ela, Um sistema composto de sensores, capaz de monitorar em tempo real o processo de tratamento da água, irá colaborar com a precisão e o resultado esperado utilizando a tecnologia do plasma frio?, conclui-se que a possibilidade de obter-se os dados tempo real, reduz alterações nos dados causados pela instabilidade nos processos do plasma, bem como a exatidão dos resultados



obtidos simultaneamente, ou seja, coletar em conjunto valores de pH, umidade, temperatura, pressão e frequência. Tendo em vista estes benefícios, garantiu que a água em tratamento não seja contaminada com outras bactérias.

Os dados obtidos pelos sensores utilizados vão de encontro ao esperado, diante da atmosfera do plasma, tornando os resultados ainda mais confiáveis.

A interface gráfica que permite a interação do usuário e a possibilidade de armazenar os resultados das coletas, contribuirá para o melhoramento do processo, bem como permitir o desenvolvimento de relatórios e gráficos para comparações futuras.

O protótipo desenvolvido adequou-se com facilidade aos equipamentos utilizados anteriormente pelo laboratório para o processo de remediação.

A avaliação do projeto realizada pela Professora Anelise, ressalta o atingimento dos objetivos bem como a solução para o problema relatado, desta forma, é observado que o sistema é eficaz, contribui de forma prática com os trabalhos realizados em laboratório e possibilitará o desenvolvimento de trabalhos futuros.

## 6.2. TRABALHOS FUTUROS

No decorrer do desenvolvimento deste trabalho, com as pesquisas e testes realizados no laboratório, bem como opiniões da professora responsável, foi observado que alguns recursos e aprimoramentos podem ser implementados como trabalhos futuros.

- Desenvolver outros protótipos do mesmo projeto, para todos os reatores de plasma do laboratório;
- Adicionar novos sensores a fim de obter-se outros dados além dos existentes, em simultâneo;
- Aprimorar ainda mais a interface gráfica a fim de tornar cada vez mais prático o uso do sistema;
- Implementar um gerador de gráficos de acordo com as necessidades do laboratório;
- Implantar o sistema web desenvolvido, em servidores na nuvem, a fim de controlar remotamente o processo de remediação, bem como, acesso a resultados salvos;

- Possibilidade de controlar um ou mais hardwares em simultâneo, através da interface gráfica, sendo local e remotamente;

## REFERÊNCIAS

ALBERING, H. J. et al. Human health risk assessment: A case study involving heavy metal soil contamination after the flooding of the river Meuse during the winter of 1993-1994. **Environmental Health Perspectives**, v. 107, n. 1, p. 37-43, 1999.

ALEXANDRE, N. Z. Influência da mineração de carvão na qualidade das águas superficiais – Revisão Bibliográfica. **Revista Tecnologia e Ambiente**, v. 2, n. 1, p. 53-61, 1996.

ALMEIDA, Felipe. **Guia da Modelagem de Dados: introdução & modelo conceitual**. Disponível em: <<https://medium.com/@felipeozalmeida/guia-da-modelagem-de-dados-introdução-modelo-conceitual-238c1f8be48>>, 2018. Acesso em: 06 de maio de 2020.

ARDUINO. Learning Arduino, 2012. Disponível em: <<http://arduino.cc/en/Tutorial/HomePage>>. Acesso em: 14 agosto de 2019.

ARAMBASIC, M. B.; BJELIC, S.; SUBAKOV, G. Acute toxicity of heavy metals (copper, lead, zinc), phenol and sodium on *Allium cepa* L., *Lepidium sativum* L. and *Daphnia magna* St.: comparative investigation and the practical applications. **Water Research**, v. 29, n. 2, p. 497-503, 1995.

ARIAS, A. R. L. et al. Utilização de bioindicadores na avaliação de impacto e no monitoramento da contaminação de rios e córregos por agrotóxicos. **Ciência & Saúde Coletiva**, v. 12, n. 1, p. 61-72, 2007.

VENTURA, PLÍNIO. **Entendendo o Diagrama de Atividades da UML**. Disponível em: <<http://www.ateomomento.com.br/uml-digrama-de-atividades/>>, 2016. Acesso em: 30 de abril de 2020.

AKCIL, A.; KOLDAS, S. Acid Mine Drainage (AMD): causes, treatment and case studies. **Journal of Cleaner Production**, v. 14, n.1 2-13, p. 1139-1145, 2006.

BASHER, A. Inhibitory effect of double atmospheric pressure argon cold plasma on spores and mycotoxin production of *Aspergillus niger* contaminating date palm fruits, **J. Sci. Food Agri**, 2015.

BANKS, S. B.; BANKS, D. Abandoned mines drainage: impact assessment and mitigation of discharges from coal mines in the UK. **Engineering Geology**, v. 60, n. 1-4, p. 31-37, 2001.

BARROS, Edna, CAVALCANTE, Sérgio. **Introdução aos Sistemas Embarcados**. Artigo Científico - Universidade Federal de Pernambuco, Pernambuco, 2011.

BENASSI, J. C et al. Evaluation of remediation of coal mining wastewater by chitosan microspheres using biomarkers. **Archives of environmental contamination and toxicology**, v. 51, n. 4, p. 633-640, 2006.

BENASSI, J. C. **O uso de bioindicadores e biomarcadores na avaliação do processo de remediação de percolado de lixiviação de carvão mineral utilizando microesferas de**

**quitosana**. 106 f. 2004. Dissertação (Mestrado) - Universidade Federal De Santa Catarina, Florianópolis, 2004.

BELL, F. G.; BULLOCK, S. E. T. The problem of acid mine drainage, with an illustrative case history. **Environmental and Engineering Geoscience**, v. 2, n. 3, p. 369-392, 1996.

BENETOLI, L.O.B. et al. Pyrite-enhanced methylene blue degradation in non-thermal plasma water treatment reactor, **J. Haz. Mat**, v. 37, p. 237-238 e p. 55-62, 2012.

BRUGGEMAN, P. J. et al. Plasma-liquid interactions: a review and roadmap, **Plasma Sources Sci. Technol**, v. 25, p. 5, 2016.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. São Paulo: Brooklin, 2007.

CAMPOS, M. L; ALMEIDA, J. A; SOUZA, L. S. Avaliação de três áreas de solo construído após mineração de carvão a céu aberto em Lauro Müller, Santa Catarina. **Revista Brasileira de Ciências do Solo**, v. 27, p. 1-6, 2003.

CHANDRA, S. et al. Comparative biomonitoring of leachates from hazardous solid waste of two industries using Allium test. **Science of Total Environment**, v. 347, p. 46-52, 2005.

CHANDANA, L. et al. Non-thermal atmospheric pressure plasma jet for the bacterial inactivation in an aqueous medium, **Sci. Total Environ**. v. 41, p. 640-641 e p. 493-500, 2018.

CODIFICAR. **O que são Requisitos Funcionais e Requisitos não Funcionais?** Disponível em: <<http://codificar.com.br/requisitos-funcionais-nao-funcionais/>>. Acesso em: 30 de abril de 2020.

CARDWELL, A.; HAWKER, D.; GREENWAY, M. Metal accumulation in aquatic macrophytes from southeast Queensland, Australia. **Chemosphere**, v. 48, n. 7, p. 653-663, 2002.

CUBAS, A. L. V.; MACHADO, M. M.; PINTO, C. R. S. C.; MOECKE, E. H. S.; DUTRA, A. R. A. Biodiesel production using fatty acids from food industry waste using corona discharge plasma technology. **Waste Management (Elmsford)**, v. 47, p. 149-154, 2015.

DALFOVO, S. Michael et al. Métodos quantitativos e qualitativos: um resgate teórico. **Revista Interdisciplinar Científica Aplicada**, v. 2, n. 4. p. 01-13, Sem. II ISSN 1980-7031, 2008

DAUBERT, L. N., BRENNAN, R. A. Passive Remediation of Acid Mine Drainage Using Crab Shell Chitin. **Environmental Engineering Science**, v. 24, p. 1475-1480, 2007.

DEVMEDIA. **O que é um UML e Diagramas de Caso de Uso: Introdução Prática a UML**. Disponível em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>, 2012. Acesso em: 29 de abril de 2020.

DEVMEDIA. **Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER)**. Disponível em: <<https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>, 2004. Acesso em: 06 de maio de 2020.

EHLBECK, J. et al. Low temperature atmospheric pressure plasma sources for microbial decontamination, **J. Phys. D Appl. Phys**, v. 44, 2015.

FALBO, R. **Engenharia de Requisitos**. p. 1–33, 2013. Disponível em: <<http://cee.uma.pt/edu/er/slides/ER-ConceitosBase.pdf>>. Citado na página 27. Acesso em: 02 de dezembro de 2019.

FISKESJO, G. The Allium test as a standard in environmental monitoring. **Hereditas**, v. 102, p. 99-112, 1985.

FOWLER, Martin. **Microservice Trade-Offs**. Disponível em: <<http://martinfowler.com/articles/microservice-trade-offs.html#summary>>. Acesso em: 19 mar. 2017.

FRIDMAN, A. **Plasma Chemistry**. New York: Cambridge University Press, 2008.

GERHARDT, A. et al. Macroinvertebrate response to acid mine drainage: community metrics and on-line behavioural toxicity bioassay. **Environmental Pollution**, v.130, n.2, p.263–274, 2004.

GEREMIAS, R. et al. Use of Biomarkers of Oxidative Stress and Genotoxicity in Allium cepa for Evaluation of Aquatic Pollution. In: XXXV REUNIÃO ANUAL DA SBBQ, 2006, Águas de Lindóia. **Caderno de Resumos da XXXV Reunião Anual da SBBQ**. São Paulo, 2006.

GORBANEV, Y. et al. Non-thermal plasma in contact with water: the origin of species, **Chemistry**, v. 22, p. 3496–3505, 2016.

HOLDINGS, A. **Rapid prototyping for microcontrollers**, 2011. Disponível em: <<http://www.mbed.org>>. Acesso em: 14 agosto de 2019.

JOHNSON, D. B.; HALLBERG, K. B. Acid mine drainage remediation options: a review. **Science of the Total Environment**, v. 338, n. 1-2, p. 3-14, 2005

KEENER, K. M. Et al. Bacterial inactivation by high- voltage atmospheric cold plasma: influence of process parameters and effects on cell leakage and DNA, **J. Appl. Microbiol**, v. 116 p. 784–794, 2014.

KOTONYA, P.; SOMMERVILLE, I. **Requirements Engineering: Processes and Techniques**. Wiley 1998.

LACLAIRE, C. Et al. Fungal decontamination by cold plasma: an innovating process for wood treatment, **Braga: COST Action IE0601/ESWM – International Conference**, pp. 5–7, 2008.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. Porto Alegre: Bookman, 2007.

LAROUSSI, M. Sterilization of contaminated matter with an atmospheric pressure plasma, **IEEE Trans. Plasma Sci**, v. 24, p. 1188–1191, 1996.

LAUS, R et al. Chitosan microspheres crosslinked with tripolyphosphate used for the removal of the acidity, iron (III) and manganese (II) in water contaminated in coal mining. **Química Nova**, v. 29, n. 1, p. 34-39, 2006.

LUKES, P. et al. Membrane damage and active but nonculturable state in liquid cultures of *Escherichia coli* treated with an atmospheric pressure plasma jet, **Bioelectrochemistry**, v. 103 p. 7, 2015.

MARCONI, A. Marina, LAKATOS, M. Eva. **Técnicas de Pesquisa**, 5 edição, Atlas, São Paulo, 2002.

MAGUREANU, M. et al. Degradation of antibiotics in water by non-thermal plasma treatment, **Water Res**, v. 45, p. 3407, 2011.

MOORE, G. Cramming more components onto integrated circuits, **Proceedings of the IEEE**, v. 86, p. 82-85, 1998.

NIETO, J. M. et al. Acid mine drainage pollution in the Tinto and Odiel rivers (Iberian Pyrite Belt, SW Spain) and bioavailability of the transported metals to the Huelva Estuary. **Environment International**, v. 33, n. 4, p. 445-455, 2007.

NUNES, B. S. et al. Use of the genus *Artemia* in ecotoxicity testing. **Environmental Pollution**, v. 144, n. 2, p. 453-462, 2006.

ORACLE. **Banco de Dados**. Disponível em: <<https://www.oracle.com/br/database/whats-is-database.html>>, 2020. Acesso em: 06 de maio de 2020.

ORTIZ, L.; Teixeira, E. C. **Meio Ambiente e Carvão: Impactos da exploração e utilização**. Porto Alegre: FEPAM, 2002.

PRESSMAN, R.S.. **Engenharia de Software**. 6o Edição McGraw-Hill, São Paulo, 2006.

RANDANI, N. et al. Ozonation of wastewater by dielectric barrier discharge and its impact on the elimination of fecal bacteria and inorganic micropollutants, **J. Eco. Heal. Env**, v. 3 p. 81, 2015.

RICHARDSON, J. Roberto. **Pesquisa Social: métodos e técnicas**, Atlas, São Paulo, 1989.

RODRIGUES, Juliana. **Conheça a importância da exploração de minérios em Santa Catarina**. 2019. Disponível em: <<http://www.floripaneews.com.br/noticia/17793-conheca-a-importancia-da-exploracao-de-minerios-em-santa-catarina>>. Acesso em: 14 agosto de 2019.

RODRIGUES, C. William. **Metodologia Científica**. Material Técnico - FEATECIST, Paracambi, 2007.

SHEORAN, A. S.; SHEORAN, V. Heavy metal removal mechanism of acid mine drainage in wetlands: A critical review. **Minerals Engineering**, v. 19, n. 2, p. 105-116, 2006.

SIGNIFICADOS. **Significado de Hardware**, 2017. Disponível em: <<https://www.significados.com.br/hardware/>>. Acesso em: 25 abril de 2020.

SILVA, A. C.; SANTANNA JR., G. L.; DEZOTTI, M. Treatment and Detoxification of a Sanitary Landfill Leachate. **Chemosphere**, v. 55, n. 2, p. 207-214, 2004.

SINONIMOS. **Sinônimo de atividade**. Disponível em: <https://www.sinonimos.com.br/atividade>. Acesso em: 30 de abril de 2020.

SMAKA-KINCL, V. et al. The evaluation of waste and ground water quality using the Allium test procedure. **Mutation Research**, v. 368, p. 171-179, 1996.

SOMMERVILLE, I. **Engenharia de Software**. 6. Ed. Editora Addison-Wesley, 2003.

SOUZA, S. M. A. G. U.; FORGIARINI, E.; SOUZA, A. A. U. Toxicity of textile dyes and their degradation by the enzyme horseradish peroxidase (HRP). **Journal of Hazardous Materials**, v.147, n. 3, p. 1073-1078, 2007.

SVENSSON, B. M. et al. Artemia salina as test organism for assessment of acute toxicity of leachatewater from landfills. **Environmental Monitoring And Assessment**, v. 102, n. 1, p. 309-321, 2005.

THOMAZINI, Daniel et al. **Sensores Industriais: Fundamentos e Aplicações**, 5 edição, p. 222, 2005, São Paulo.

THOPAN, P. Et al. Q. Huang, Effect of N<sub>2</sub>/ O<sub>2</sub> composition on inactivation efficiency of Escherichia coli by discharge plasma at the gas-solution interface, Clin. **Plasma Med.** 7, p. 1–8, 2017.

TOCCI, R. J. et al. Sistemas Digitais: princípios e aplicações, **Person**, 15 Edição, 2011.

UNIVERSIDADE DO SUL DE SANTA CATARINA. Pró-Reitoria Acadêmica. Programa de Bibliotecas. **Trabalhos acadêmicos na Unisul**: apresentação gráfica para TCC, monografia, dissertação e tese. Cristiane Salvan Machado et. al. (Org). 2. ed. rev. e ampl. Tubarão : Ed. Unisul, 2008.

VIEIRA, JOSE NEWTON et al. **Introdução aos fundamentos da computação**, 1 edição, p. 29, 30. Cap. 2, Máquinas de estados finitos, Editora Thomson, 2004.

WENDLING, MARCELO. **Sensores**. Material Técnico Versão 2.0 - Universidade Estadual Paulista - Colégio Técnico Industrial de Guaratinguetá Professor Carlos Augusto Patrício Amorim, Guaratinguetá, 2010.

## APÊNDICES

### APÊNDICE A – Cronograma I

O presente cronograma considera o prazo da concessão de 10 (dez) meses. As atividades estão distribuídas de acordo com as demandas alinhadas em aula e com o Professor Orientador.

Atividades	Agosto	Setembro	Outubro	Novembro	Dezembro
<i>Capítulo 1</i>	X				
<i>Capítulo 2</i>		X	X		
<i>Capítulo 3</i>			X	X	
<i>Capítulo 4</i>					X

Atividades	Fevereiro	Março	Abril	Maio	Junho
<i>Capítulo 4</i>	X				
<i>Capítulo 5</i>		X	X		
<i>Capítulo 6</i>				X	X

### APÊNDICE B – Autorização para uso de nome e respostas de avaliação do projeto

Eu, Anelise Leal Vieira Cubas, autorizo os autores do presente trabalho, a citar meu nome, bem como as respostas sobre a avaliação dos resultados obtidos com o desenvolvimento do projeto.

---

Professora Dra. Anelise Leal Vieira Cubas  
Universidade do Sul de Santa Catarina