



UNIVERSIDADE ESTADUAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
MESTRADO EM CIÊNCIA E TECNOLOGIA AMBIENTAL

THALES LACERDA QUERINO DE ALBUQUERQUE

AUTOMAÇÃO DE BIORREATORES DE MEMBRANA UTILIZANDO A
PLATAFORMA ARDUINO

Campina Grande

2017

THALES LACERDA QUERINO DE ALBUQUERQUE

AUTOMAÇÃO DE BIORREATORES DE MEMBRANA UTILIZANDO A
PLATAFORMA ARDUINO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia Ambiental da Universidade Estadual da Paraíba, em cumprimento as exigências para obtenção do título de mestre.

Orientador: Prof. Dr. Wilton Silva Lopes

Campina Grande

2017

É expressamente proibida a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano da dissertação.

A345a Albuquerque, Thales Lacerda Querino de.
Automação de biorreatores de membrana utilizando a plataforma arduino [manuscrito] / Thales Lacerda Querino de Albuquerque. - 2017.
57 p. : il. color.

Digitado.
Dissertação (Mestrado Ciência e Tecnologia Ambiental) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2017.
"Orientação: Prof. Dr. Wilton Silva Lopes, Departamento de Engenharia Sanitária Ambiental".

1. Lodo de esgoto. 2. Biorreatores de membrana. 3. Arduino. 4. Software. I. Título.

21. ed. CDD 005.3

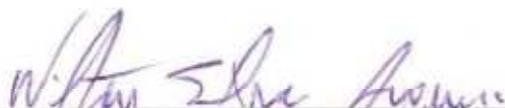
THALES LACERDA QUERINO DE ALBUQUERQUE

**AUTOMAÇÃO DE BIORREATORES DE MEMBRANA UTILIZANDO A
PLATAFORMA ARDUINO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia Ambiental da Universidade Estadual da Paraíba, em cumprimento as exigências para obtenção do título de mestre.

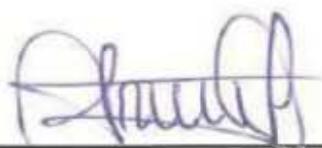
Aprovado em 17 de fevereiro de 2017.

BANCA EXAMINADORA:



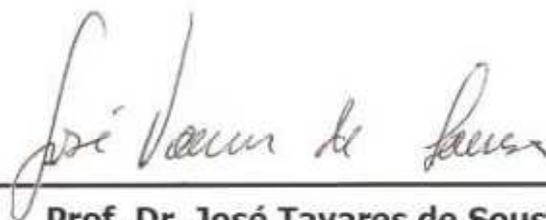
Prof. Dr. WILTON SILVA LOPES

(Orientador – Universidade Estadual da Paraíba)



Prof. Dr. Roger Ruben Huaman Huanca

(Examinador Externo – Universidade Estadual Paulista - UNESP)



Prof. Dr. José Tavares de Sousa

(Examinador Interno – Universidade Estadual Paulista - UNESP)

AGRADECIMENTOS

À Deus, pela vida, saúde, coragem, força de vontade e por me fazer superar os obstáculos que surgiram na minha vida, com suas bênçãos e abundância de graças.

Aos meus pais, por me incentivarem nos estudos e por toda educação e ensinamentos que me deram durante a vida.

A minha esposa por sempre estar ao meu lado, me dando amor e forças para meu crescimento pessoal e profissional.

Ao orientador Wilton Silva Lopes pela oportunidade, incentivo e amadurecimento dos meus conhecimentos e conceitos que me levaram a execução e conclusão desta dissertação de mestrado.

A todos os colegas de laboratório que de alguma forma contribuíram na conclusão desta pesquisa.

A todos os professores envolvidos nesta pesquisa, que ao longo desta jornada me deram conselhos valiosos para alcançar o sucesso na pesquisa. Em especial aos professores participantes da banca Prof. Dr. Roger Ruben Huaman Huanca e ao Prof. Dr. José Tavares de Sousa.

Minha gratidão a todos.

RESUMO

Com o crescimento das redes de coleta de esgotos, a produção de lodos em Estações de Tratamento de Esgotos (ETEs) vêm crescendo continuamente à medida que estas redes são ampliadas e novas estações de tratamento são implantadas. O lodo gerado nas ETEs apresenta uma composição de constituintes que podem trazer riscos para a saúde da população e ao meio ambiente, sendo necessário realizar disposição final adequada a este resíduo ou reaproveitar os nutrientes que podem ser obtidos em processos de filtração por membranas. Estudos recentes demonstram que a recuperação de nutrientes do lodo de esgoto pode ser realizada em sistemas de Biorreatores Anaeróbios de Membrana - AnMBR, que incluem um reservatório de alimentação, um digestor anaeróbio, um módulo de filtração de membrana e um reservatório de armazenamento de permeado. Alguns fatores contribuem no desempenho do processo do biorreator, tornando necessário o acompanhamento constante de alguns parâmetros. O acompanhamento constante foi realizado através da automação do sistema, utilizando a placa micro controladora Arduino, conectada a sensores de pressão e temperatura, onde os valores são exibidos através de relatórios gerados pelo software. O software mostra informações em tempo real, como também disponibiliza uma ferramenta onde é possível consultar dados do biorreator de acordo com a data desejada. Ao término da pesquisa foi possível constatar que o software facilita a armazenagem e tratamento de dados, oferecendo um melhor gerenciamento das informações de extrema importância no auxílio à tomada de decisão na gestão de biorreatores de membrana.

Palavras chave: Lodos de esgoto, biorreatores de membrana, automação, Arduino, software.

ABSTRACT

With the growth of sewage collection networks, sludge production in Sewage Treatment Plants (ETEs) has been growing steadily as these networks are expanded and new treatment plants are deployed. The sludge generated in the ETEs presents a composition of constituents that can pose risks to the health of the population and the environment, and it is necessary to make adequate final disposal of this residue or to reuse the nutrients that can be obtained in membrane filtration processes. Recent studies have shown that nutrient recovery from sewage sludge can be performed in Anaerobic Membrane - AnMBR systems, which include a feed reservoir, an anaerobic digester, a membrane filtration module and a permeate storage reservoir. Some factors contribute to the performance of the bioreactor process, making constant monitoring of some parameters necessary. The constant monitoring was performed through the automation of the system, using the micro controller Arduino board, connected to pressure and temperature sensors, where values are displayed through reports generated by the software. The software shows information in real time, but also provides a tool where you can query data from the bioreactor according to the desired date. At the end of the research it was possible to verify that the software facilitates the storage and processing of data, offering a better management of information of extreme importance in the aid of decision making in the management of membrane bioreactors.

Keywords: Sewage sludge, membrane bioreactors, automation, Arduino, software.

LISTA DE FIGURAS

Figura 1 - Características dos principais processos de separação por membranas	14
Figura 2 - (A) Placa Arduino Uno e (B) detalhe dos componentes da placa. ...	16
Figura 3 - Sistema em funcionamento com resultados visualizados em tempo real.....	25
Figura 4 - (A) Membrana com sensores de pressão instalados, (B) Membrana com lodo em circulação	26
Figura 5 - Esquema de conexão física entre Biorreator, Módulo de Membrana, Computador servidor, Arduino e Sensores.....	28
Figura 6 - Conexão entre as partes que envolvem o sistema de informação. .	30
Figura 7 - Código SQL com a estrutura de criação das tabelas do banco de dados	32
Figura 8 - Estrutura das tabelas do sistema	33
Figura 9 - Estrutura de diretórios e arquivos do software AnMBR Web.....	34
Figura 10 - Sensor de Pressão instalado no biorreator para medir a produção de Biogás.....	37
Figura 11 - Menu de acesso do software AnMBR	38
Figura 12 - Página inicial do software	39
Figura 13 - Página Configurações.....	40
Figura 14 - Módulo Relatório Produção de Biogás.	42
Figura 15 - Módulo relatório Pressão Transmembrana.	44
Figura 16 - Relatório Temperatura	46
Figura 17 - Gráfico Pressão Transmembrana	47

LISTA DE ABREVEATURAS E NOMECLATURAS

AnMBR - Biorreatores Anaeróbios de Membrana

ASP - Active Server Pages

CSS - Cascading Style Sheets (Folhas de Estilo em Cascata)

ETEs - Estações de Tratamento de Esgotos

GND – “Ground” – Pino de referência terra, ausência de tensão ou 0 volts.

HTML - HyperText Markup Language (Linguagem de Marcação de Hipertexto)

HTTP - Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)

IIS - Internet Information Services

JS – Java Script

Kpa – Unidade de pressão, chamada QuiloPascal.

LCD – Liquid Crystal Display (Tela de Cristal Líquido)

PHP - Hypertext Preprocessor (Pré-processador de hipertexto)

Psi – Unidade de medida de pressão no sistema inglês.

SQL – Structured Query Language (Linguagem de Consulta Estruturada)

ODBC - Open Database Connectivity

SGBD – Sistema Gerenciador de Banco de Dados

VCC – Tensão corrente contínua

Web – Rede de alcance mundial

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVOS	12
2.1	Geral.....	12
2.2	Específicos	12
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	Separação por membranas.....	13
3.2	Arduino	14
3.3	O microcontrolador ATmega328 do Arduino Uno.....	15
3.4	Shields e sensores para Arduino	17
3.4.1	Sensor de Pressão MPX4250	18
3.4.2	Sensor de Temperatura DS18B20	18
3.5	Servidores WEB.....	19
3.5.1	Linguagem de Marcação.....	21
3.5.2	Linguagem de programação PHP	22
3.5.3	Banco de Dados MySQL.....	23
4	MATERIAL E MÉTODOS	24
4.1	Considerações gerais	24
4.2	Configurações do Biorreator de membrana	25
4.3	Automação do Biorreator com a plataforma Arduino	27
4.4	Conexão entre Arduino e sensores	28
4.5	Estruturação do sistema de informação	29
4.6	Desenvolvimento do Software Web	30
4.6.1	Equipamentos	30
4.6.2	Linguagem computacional	31
4.6.3	Bando de Dados	31
4.6.4	Frameworks	33
4.7	Estrutura do software AnMBR Web.....	34
4.8	Metodologia desenvolvida para cálculo da produção de biogás.....	36
5	RESULTADOS E DISCUSSÕES.....	38
5.1	Descrição do sistema.....	38
5.2	Página inicial.....	39
5.3	Página configurações	39

5.4	Módulo relatórios	40
5.4.1	Produção de biogás	40
5.4.2	Pressão transmembrana	43
5.4.3	Temperatura	44
5.4.4	Gráfico Pressão Transmembrana	46
6	CONCLUSÕES	49
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	50
8	APÊNDICE A – Página inicial do servidor Web	52
9	APÊNDICE B – Arquivo de importação com todas as classes utilizadas ..	54
10	APÊNDICE C – Classe Conexao.php.....	54
11	APÊNDICE D - Classe Comando.php	54
12	APÊNDICE E – Classe Pressao.php	55
13	APÊNDICE F – Classe PressaoBiogas.php	56
14	APÊNDICE G – Classe Temperatura.php	57

1 INTRODUÇÃO

Com o crescimento das redes de coleta de esgotos, a produção de lodos em Estações de Tratamento de Esgotos (ETEs) vêm crescendo continuamente à medida que estas redes são ampliadas e novas estações de tratamento são implantadas.

O lodo gerado nas ETEs apresenta uma composição de constituintes que podem trazer riscos para a saúde da população e ao meio ambiente, sendo necessário realizar disposição final adequada a este resíduo ou reaproveitar os nutrientes que podem ser obtidos em processos de filtração por membranas.

A correta gestão do lodo é, portanto, um problema ambiental e sanitário relevante para as empresas de saneamento públicas ou privadas que devem seguir as recomendações contidas na legislação ambiental que institui a Política Nacional de Resíduos Sólidos, que no caso do Brasil, refere-se à Lei 12.305/2010.

O lodo de esgoto produzido nas ETE's não representa apenas problemas, mas seu tratamento adequado pode trazer diversos benefícios. O lodo tem sido valorizado, devido à possibilidade de produzir biogás, podendo ser utilizado como fonte de energia. Principalmente em função da existência de biomassa tratada por meio do processo de biorreação anaeróbia, a produção de biogás tem mostrado cada vez mais atrativa, devido à geração de biogás com alto teor de metano (CH₄).

Outra possibilidade de reutilização do lodo de esgoto está relacionada ao seu uso como nutrientes na agricultura, como o nitrogênio, fósforo e enxofre presentes em quantidades significativas que podem ser recuperados através de sistemas adequados. No entanto, Gerardo *et al.* (2013) relatam que a remoção e recuperação de nutrientes do lodo de esgoto não é uma tarefa fácil.

Estudos recentes demonstram que a recuperação de nutrientes do lodo de esgoto pode ser realizada em sistemas de Biorreatores Anaeróbios de Membrana - AnMBR (BELLI *et al.*, 2012, p. 144). Estes sistemas incluem um reservatório de alimentação, um digestor anaeróbio, um módulo de filtração de membrana e um reservatório de armazenamento de permeado.

Alguns fatores podem contribuir para o desempenho do processo do biorreator, tornando fundamental o monitoramento contínuo dos parâmetros

operacionais, como pressão transmembrana e temperatura do líquido no biorreator.

Os parâmetros operacionais são obtidos através de sensores, estes são conectados com uma placa micro controladora tornando-se capaz de enviar as informações obtidas para armazenamento em uma base de dados. No entanto, torna-se necessário o uso de tecnologia uma que forneça uma alternativa viável para realização desta tarefa.

A tecnologia escolhida para automação do projeto foi Arduino. Trata-se de uma plataforma de hardware livre, projetada com um microcontrolador com suporte de entrada/saída embutido e uma linguagem de programação personalizada. O objetivo do Arduino é permitir a criação de ferramentas que sejam acessíveis e de baixo custo (EVANS; NOBLE; HOCHENBAUM, 2013, p. 25).

Esta pesquisa viabilizará a utilização de mecanismos de baixo custo para instalar e configurar os sensores e placas no biorreator de membrana, bem como disponibilizará um software Web que facilita a inserção e consulta dos dados obtidos através dos sensores e placas instalados no biorreator.

O sistema desenvolvido mostra os dados gerados através de Web páginas e gráficos que facilita a interpretação do que está acontecendo no processo de recuperação de nutrientes, para que o operador do sistema possa tomar as decisões que julgar mais adequadas à correção dos parâmetros relacionados à recuperação de nutrientes.

2 OBJETIVOS

2.1 Geral

Automatizar um sistema de biorreatores utilizando uma placa micro controladora Arduino acoplada a sensores de pressão e temperatura.

2.2 Específicos

- Instalar um sensor no biorreator para medir a pressão exercida sobre este devido a mistura de gases do biogás;
- Instalar sensores de pressão no módulo de membrana para monitorar os três diferentes pontos: entrada do lodo no módulo de membrana, saída do lodo para recirculação no biorreator e saída do permeado;
- Desenvolver um software Web que facilite a análise dos dados obtidos através dos sensores.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Separação por membranas

A tecnologia de separação por membranas surgiu no início dos anos 60, sendo primeiramente comercializada para fins de dessalinização de água do mar, na forma de sistemas de osmose reversa. Na década de 80, foi desenvolvida uma alternativa derivada desta tecnologia, a nanofiltração, sendo instalada em escala comercial, na Noruega, para remoção de cor em águas derivadas de zonas com turfas, e no estado da Flórida – EUA, para remoção de dureza de águas subterrâneas (MAESTRI, 2007).

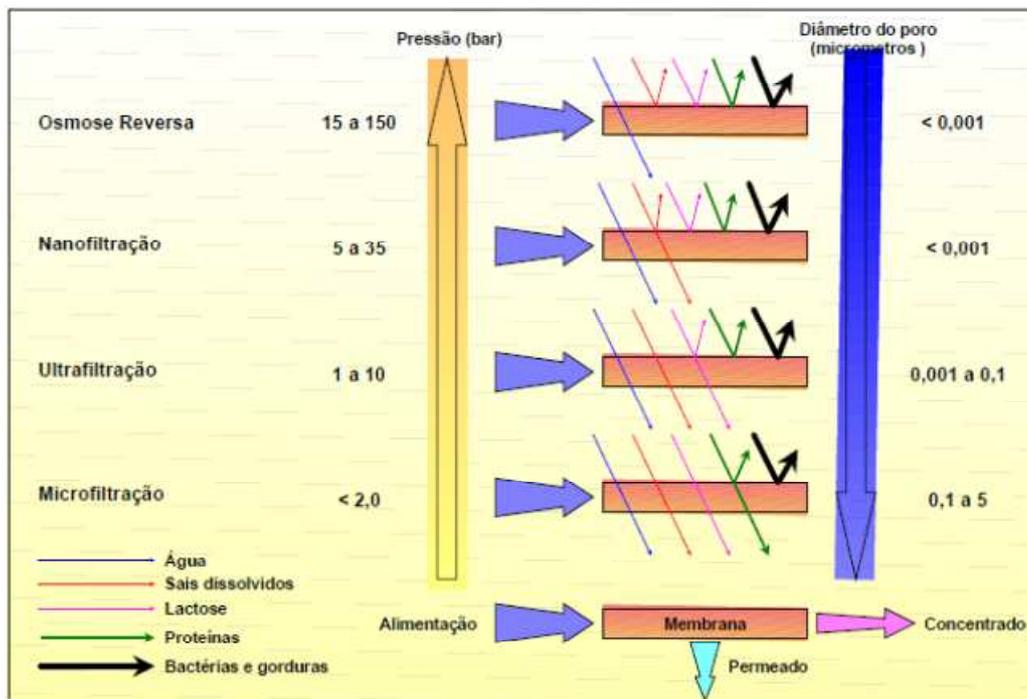
Estes dois segmentos de mercado, entretanto, representam pequenos nichos dentro do universo dos sistemas utilizados no tratamento de água e esgoto em saneamento básico. O uso de membranas vem crescendo de forma significativa em todo o mundo, inclusive no Brasil. Atualmente, esta tecnologia vem sendo muito utilizada, principalmente para o tratamento de águas residuais (SCHNEIDER; TSUTIYA, 2001).

Membranas são barreiras que tem a função de reter fisicamente e/ou quimicamente as partículas presente num fluido, permitindo a passagem de algumas espécies e a retenção de outras. Segundo Viana (2004), uma membrana consiste em um filme que separa duas fases, agindo como uma barreira semipermeável e seletiva, restringindo total ou parcialmente o transporte de uma ou várias espécies químicas presentes nas soluções.

Os principais processos de separação por membranas para o tratamento de água são a microfiltração, ultrafiltração, nanofiltração e a osmose reversa que utilizam pressão hidráulica como força motriz para separar a água dos contaminantes (MIERZWA et al., 2008a).

Como ilustrado na figura 1, cada classificação opera com distintas pressões, no processo de osmose reversa existe um maior consumo energético, dado que este trabalha com pressões elevadas, no entanto, este processo apresenta a maior capacidade de separação de contaminantes. O diâmetro representa a abertura dos poros das membranas, sendo o principal fator na retenção de poluente.

Figura 1 - Características dos principais processos de separação por membranas



Fonte: MIERZWA et al., 2008a

Em função da aplicação a que se destinam, as membranas apresentam diferentes estruturas, podendo ser categorizadas em densas e porosas. Ambas podem ainda ser classificadas como isotrópicas e anisotrópicas (PROVENZI, 2005). São denominadas isotrópicas membranas com mesmas características morfológicas e anisotrópicas ou assimétricas, membranas com características diferentes.

De acordo com sua natureza, as membranas podem ser classificadas como sintéticas ou biológicas, sendo as membranas biológicas essenciais para a manutenção da integridade física das células, enquanto que as membranas sintéticas são fabricadas e podem ser de natureza orgânica ou inorgânica (DE SOUSA, 2008).

3.2 Arduino

O Arduino foi originalmente projetado como um recurso para auxiliar os estudantes no ensino, mas em 2005 ele foi comercialmente lançado por Massimo Banzi e David Cuartielles, tornando-se um produto de sucesso entre

fabricantes e estudantes devido a sua fácil utilização e a durabilidade que ele proporciona (MONK, 2013).

O Arduino é um pequeno computador capaz de processar informações de dispositivos e componentes externos conectados a ele através dos pinos de entradas e saídas (14 digitais e 6 entradas analógicas) de acordo com o que é programado. Mcroberts (2011) afirma que, o Arduino é uma plataforma de computação física ou embarcada, isto é, um dispositivo de hardware com um sistema de software interagindo com o ambiente.

O Arduino é um dispositivo formado principalmente por 2 componentes básicos: a placa, que é o elemento de hardware utilizado para construir seus objetos e a IDE (*Integrated Development Environment*) do Arduino, que é um programa executado no computador por onde escrevemos o código (chamado de *sketch*) e que fará o envio para a placa Arduino (BANZI; SHILOH, 2010).

Existem diversas placas de Arduino e todas utilizam a mesma linguagem para ser programada, mas a versão UNO é a mais utilizada e a maioria das placas usam as mesmas conexões com o ambiente externo, o que permite usar qualquer modelo facilmente (MONK, 2013).

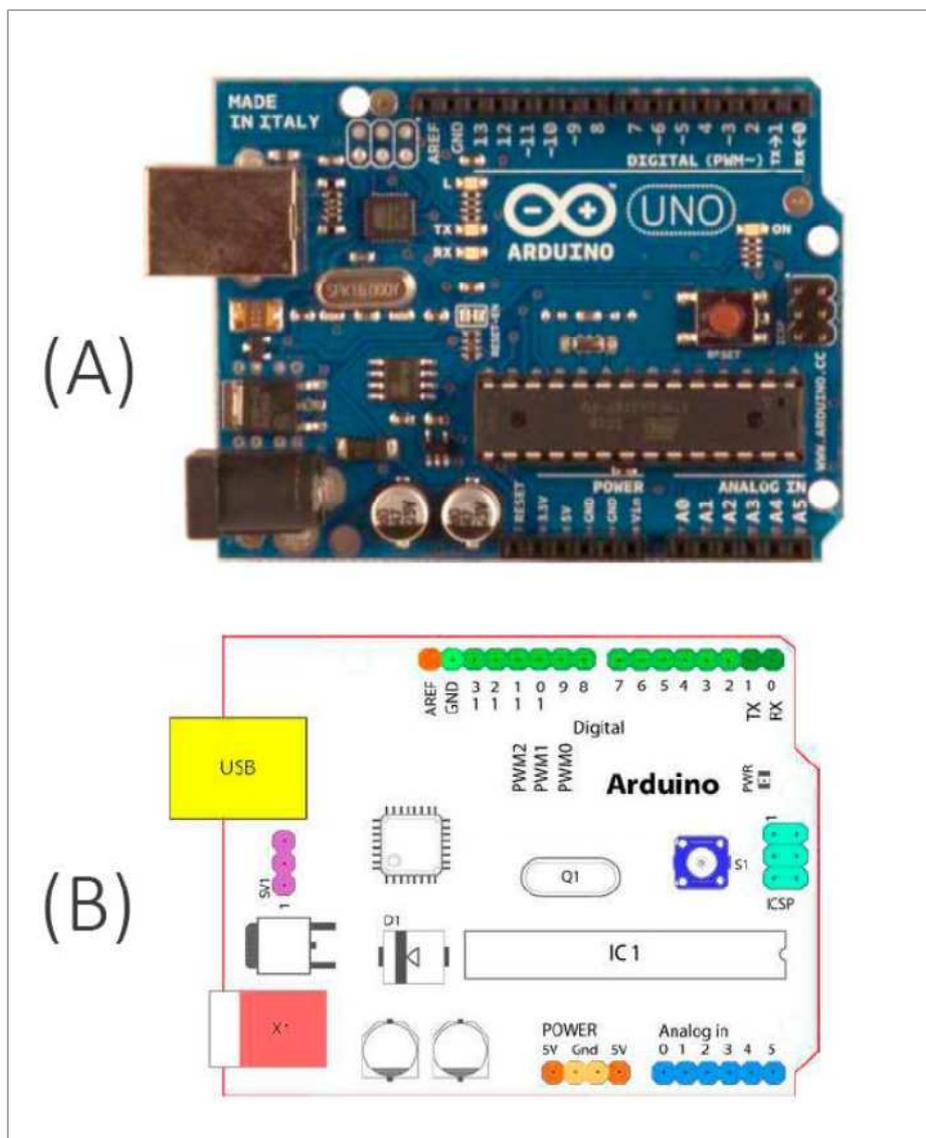
O software e o hardware do Arduino são de fonte aberta a qualquer pessoa, isso quer dizer que tanto os códigos como os esquemas e também os projetos podem ser usados livremente por qualquer um e para qualquer propósito (MCROBERTS, 2011).

3.3 O microcontrolador ATmega328 do Arduino Uno

Atualmente, há vários modelos de placas no mercado, este projeto utiliza a versão da placa conhecida como “Arduino Uno”. Este modelo de placa é constituído por um microprocessador Atmel AVR (ATmega328), um cristal ou oscilador (relógio que envia pulsos em uma frequência especificada para permitir sua operação na velocidade correta) e um regulador linear de 5 volts.

A placa Arduinio também possui seis pinos de entradas analógicas, quatorze pinos de entrada/saída digital, a placa também possui uma conexão USB para conecta-la ao computador e uma entrada de alimentação externa (FABRI JUNIOR, 2014).

Figura 2 - (A) Placa Arduino Uno e (B) detalhe dos componentes da placa.



Fonte: Adaptado de FABRI JUNIOR, 2014.

A figura 2(B) ilustra os detalhes dos componentes da placa Arduino Uno. Os detalhes dos componentes ilustrado na Figura 2(B) são descritos a seguir. Iniciando pela imagem na parte superior, no sentido horário tem-se:

- Laranja – pino analógico de referência;
- Verde claro – pino terra;
- Verde escuro – pinos digitais 2-13;

Verde escuro – pinos digitais 0-1, entrada serial, saídas TX/RX e esses pinos não podem ser utilizados como entrada/saída (digitalRead e digitalWrite), caso esteja usando comunicação serial;

Azul escuro – S1, botão de Reset;

Azul claro – pinos analógicos 0-5;

Laranja escuro e laranja claro – pinos de alimentação de 5V, ground (GND) e 3,3V;

Rosa – X1, conexão de fonte externa de alimentação 6V a 20V;

Roxo – SV1, alternado de alimentação externa e alimentação USB;

Amarelo – Entrada de conexão para USB, usado para fazer upload da programação para a placa, comunicação entre a placa e o computador, pode ser usado para alimentar a placa.

3.4 Shields e sensores para Arduino

Alguns componentes foram desenvolvidos para diferentes tipos de finalidades e projetos, vários fabricantes lançam frequentemente no mercado uma diversidade de placas expansivas (mais conhecida como Shields) e sensores, estes são projetadas para trabalhar em paralelo com placas micro controladoras em projetos de automação.

Shields ou expansores, são placas que podem ser conectadas em cima da placa Arduino, estendendo as suas capacidades, e são fixados no Arduino através de uma conexão alimentada por pinos-conectores. A capacidade de expansão possibilita uma grande quantidade de aplicações de maneira simples e rápida. Os shields seguem a mesma filosofia que o kit de ferramenta original, eles são fáceis de montar e baratos de produzir (FABRI JUNIOR, 2014).

Os circuitos contidos nos diversos shields contém uma eletrônica que adiciona funções que a placa principal não possui. Como exemplos temos shields para controle de motores, shields para comunicação com a internet, shields para acoplamento de sensores, entre outros.

3.4.1 Sensor de Pressão MPX4250

O sensor da série MPX4250 é projetado para detectar a pressão absoluta do ar, utilizando-o como altímetro ou barômetro. Este sensor da Freescale integra on-chip, amplificadores operacionais e redes de resistores para fornecer um alto nível de sinal de saída analógica e compensação de temperatura.

Este sensor opera em 5V atendendo uma faixa de 0 à 250 kPa (ou 0 à 36.3 psi). É conectado, através do pino 1, em uns dos canais analógicos do micro controlador. Os pinos 4, 5 e 6 não são utilizados. O pino 2 é o GND e o pino 3 é o VCC. (GHANI; ABDULLAH, 2014)

Conforme consulta realizada no documento, que consta todos os dados e características técnicas do equipamento, disponibilizado pelo fabricante do produto em seu site (FREESCALE, 2009), algumas características deste sensor são relacionadas a seguir:

- Aplicações diferenciais e de calibre disponíveis;
- 1,4% erro máximo acima de 0 ° a 85 ° C;
- Medidor patenteado de tensão;
- Compensação de temperatura acima de -40 ° a + 125° C;
- Oferece redução de peso e volume comparado aos módulos híbridos existentes;
- Elemento epóxi durável.

Ele é concebido para uma ampla gama de aplicações, particularmente as que utilizam um microcontrolador com entradas analógicas e digitais, o que torna simples a conexão entre o sensor e a placa Arduino.

3.4.2 Sensor de Temperatura DS18B20

A medição de temperatura no ambiente industrial engloba uma variedade ampla de necessidades e aplicações. Para atender a essa ampla gama de necessidades, a indústria de controles de processo desenvolveu um grande número de sensores e dispositivos para lidar com essa demanda.

O sensor de temperatura DS18B20 foi escolhido neste projeto devido ao seu baixo custo e biblioteca de fácil disponibilidade. O nome de biblioteca, do

inglês, library, refere-se a um conjunto de funções escritas por outros programadores que já resolvem determinados problemas para você sem que você precise “reinventar a roda”.

O DS18B20 é um sensor de temperatura que permite fazer leituras de temperaturas na faixa de 55-125 ° C, não precisa de conversor analógico para digital, o valor da temperatura é convertido diretamente para digital. Possui suporte para função de rede multiponto, o que permite que vários sensores desse modelo possam ser instalados em um mesmo projeto (LIU, 2013). Mais características deste sensor são relacionadas a seguir:

- Interface de um único fio: apenas um fio para conectar com o microcontrolador Arduino;
- A precisão da medida pode chegar a 0,5 ° C;
- As leituras de temperatura têm nove dígitos;
- O tempo de conversão analógico para digital é de 200 milissegundos;

Após serem obtidos os valores de temperatura através do sensor DS18B20, o qual está conectado a placa Arduino através de uma de suas entradas digitais, os dados são enviados para o computador servidor e armazenados no banco de dados.

3.5 Servidores WEB

Inicialmente, no surgimento da Internet, a rede, denominada World Wide Web (WWW) baseava-se em páginas estáticas, consistindo apenas em repositórios de informação (servidores) que continham documentos estáticos, sendo que a invenção dos browsers foi feita para servir de meio de envio e apresentação dos referidos documentos. A internet dos dias de hoje é incomparavelmente diferente daquilo que era na sua forma inicial. Atualmente, a maioria das páginas web assumem a forma de aplicações, sendo altamente funcionais, baseando-se numa relação intrínseca entre o web browser e o servidor (GARCIA, 2012).

O servidor Web tem grande importância na troca de informações e como essas informações se apresentam em toda a rede mundial de computadores.

Quando se acessa um site da Internet pelos navegadores disponíveis no mercado é realizada uma troca de informações entre o navegador e o servidor Web através de um protocolo chamado HTTP (HyperText Transfer Protocol, na tradução: Protocolo de Transmissão de Hipertexto).

O servidor Web será o computador responsável por armazenar os dados obtidos através dos sensores. Neste servidor torna-se necessário a instalação dos seguintes serviços:

Servidor Web Apache – Responsável por hospedar inúmeras aplicações Web;

Módulo PHP – módulo para rodar sites criados na linguagem de programação PHP;

Sistema Gerenciador de Banco de Dados (SGBD) MySQL – sua função é armazenar em tabelas todos os dados relacionados ao funcionamento do programa, estas informações envolvem desde dados dos utilizadores do software, como dados obtidos através dos sensores.

O software Web desenvolvido, escrito em linguagem de programação PHP e linguagem de marcação HTML, é composto por arquivos com extensões (.html e .php), estes arquivos são armazenados em um diretório gerenciado pelo servidor Web apache.

O Servidor Apache é um software livre, o que significa que seu código-fonte pode ser estudado e alterado, além de, poder utilizá-lo gratuitamente. É devido a essas características que o software vem sendo melhorado ao decorrer dos anos. E em muitas vezes graças ao trabalho voluntario de desenvolvedores ao redor do mundo, levando isso o Servidor Apache o servidor Web o mais usado no mundo.

Como um servidor Web, o Apache é responsável por aceitar solicitações (HTTP) do diretório de usuários da Internet e enviá-los a sua informação desejada na forma de arquivos e páginas da Web. Grande parte do software e código da Web é projetado para trabalhar junto com características Apache. Os programadores que trabalham em aplicações Web geralmente fazem uso de uma versão doméstica do Apache para pré-visualizar e testar o código. Apache também possui um recurso de compartilhamento de arquivos seguro, permitindo aos usuários colocar arquivos no diretório principal do seu software e compartilhá-los com outros usuários.

Servidor Apache é o servidor que executa os códigos HTML e PHP devolvendo a tela de interação para o usuário, ele lê uma página escrita em formato de códigos, interpreta e gera uma saída de fácil compreensão por qualquer pessoa que esteja utilizando o software.

3.5.1 Linguagem de Marcação

A homepage, também conhecida como página Web é compreendida por 'tags', que são comandos especiais, e por textos de uma linguagem de programação denominada HTML (HyperText Markup Language). A simplicidade dessa linguagem é baseada na finalidade de formatar o texto mostrado e criar relações entre outras homepages, com isso, cria-se documentos com o conceito de hipertextos.

Quando um navegador mostra uma página Web, ele analisa o texto da página e busca por caracteres especiais que são chamados de tags. Essas tags é que irão dizer como a informação contida na página deverá ser mostrada na tela para o usuário. As tags são sinais do tipo "< >" e "< />" e podem informar várias formatações ao texto, como por exemplo: se o texto será exibido em qual cor ou se será em negrito, itálico ou sublinhado, se será um endereço de outra página, entre outras coisas (RAMALHO, 1996).

A linguagem de marcação HTML tem grande importância nesta pesquisa, onde através dos códigos criados usando essa linguagem, definem-se o formato dos valores apresentados nos relatórios disponibilizados no software. Todos os relatórios disponibilizados no software utilizam um conjunto de tags HTML para exibir o resultado final esperado pelo usuário, com informações úteis e claras de forma acessível.

HTML foi criada com o intuito de apresentar o conteúdo da página, por isso pode ser chamada de linguagem de marcação ou linguagem de apresentação. Diferentemente da linguagem de marcação, a linguagem de programação é um conjunto de convenções e regras que especificam como transmitir informações entre pessoas e máquinas.

3.5.2 Linguagem de programação PHP

O PHP (Hypertext Pre-Processor) é uma linguagem de programação desenvolvida especialmente para a Web, com a possibilidade de ser embutida em código HTML, criada por Rasmus Lerdorf e mais tarde aperfeiçoada por Stig Bakken, Andi Gutmans e Zeev Suraski. O PHP é uma linguagem de servidor, (server-side), isto significa que as páginas Web são processadas no servidor, permitindo assim que alterações feitas em uma aplicação PHP sejam realizadas de forma imediata em todos os computadores clientes que acessam a página.

PHP por ser uma linguagem de script interpretada que executa no servidor www, criada exclusivamente para a Internet, tem atributos que flexibilizam o desenvolvimento de páginas dinâmicas, com saída de dados em diversos formatos como página de marcação HTML, imagens, textos formatados em PDF, geradas dinamicamente através de informações recebidas do computador-cliente, armazenados em disco ou existentes em um banco de dados (SOARES, 2007).

Uma das grandes vantagens em utilizar o PHP está no fato de ser considerada uma linguagem extremamente simples para um iniciante, no entanto oferece muitos recursos para um programador profissional. É possível fazer muita coisa com esta linguagem, como coletar dados de formulários, gerar páginas com conteúdo dinâmico, criar gráficos, planilhas, e muito mais.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes do Unix (como HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros. O PHP também é suportado pela maioria dos servidores Web atualmente. Isso inclui o Apache, o IIS e muitos outros (SOARES, 2007).

O PHP funciona com as mais populares bases de dados – permite suporte nativo para as bases de dados habitualmente usadas em aplicações Web, tais como Oracle, SQL Server e MySQL. Este suporte nativo dá ao PHP uma margem de desempenho superior a outras linguagens, como o ASP que requer ligações ODBC a muitas bases de dados.

3.5.3 Banco de Dados MySQL

Banco de dados pode ser definido como uma coleção de dados estruturados. Com ele, é possível adicionar, acessar e processar dados armazenados em um banco de dados de computador, no entanto é necessário um sistema de gerenciamento de banco de dados, como o servidor MySQL.

O MySQL é atualmente um dos bancos de dados mais populares do mundo, sendo disponibilizado de forma gratuita e multi plataforma, ou seja, pode ser executado em diversos sistemas operacionais como Windows, Linux e Unix, considerando também sua agilidade na execução das sentenças e mínima necessidade de configuração e administração.

MySQL é um SGBD (Sistema Gerenciador de Bancos de Dados) relacional que utiliza a linguagem padrão SQL (Structured Query Language) e é largamente utilizado em aplicações para a Internet. É o mais popular entre os bancos de dados com código-fonte aberto.

Sendo considerado um banco de dados relacional, o MySQL armazena dados em tabelas separadas em vez de colocar todos os dados em um só local, sendo uma alternativa atrativa porque, mesmo possuindo uma tecnologia complexa de banco de dados, seu custo é bastante baixo. Tem como destaque suas características de velocidade, escalabilidade e confiabilidade, o que vem fazendo com que ele seja adotado por departamentos de TI (Tecnologia da Informação), desenvolvedores Web e vendedores de pacotes de softwares.

A seguir são listadas algumas vantagens do MySQL:

- número ilimitado de utilização por usuários simultâneos;
- capacidade de manipulação de tabelas com mais de 50.000.000 de registros;
- alta velocidade de execução de comandos;
- fácil e eficiente controle de privilégios de usuários.

Portanto, o MySQL e o PHP formam uma excelente dupla para o desenvolvimento de páginas Web dinâmicas, tanto para Websites pequenos como para grandes portais.

4 MATERIAL E MÉTODOS

4.1 Considerações gerais

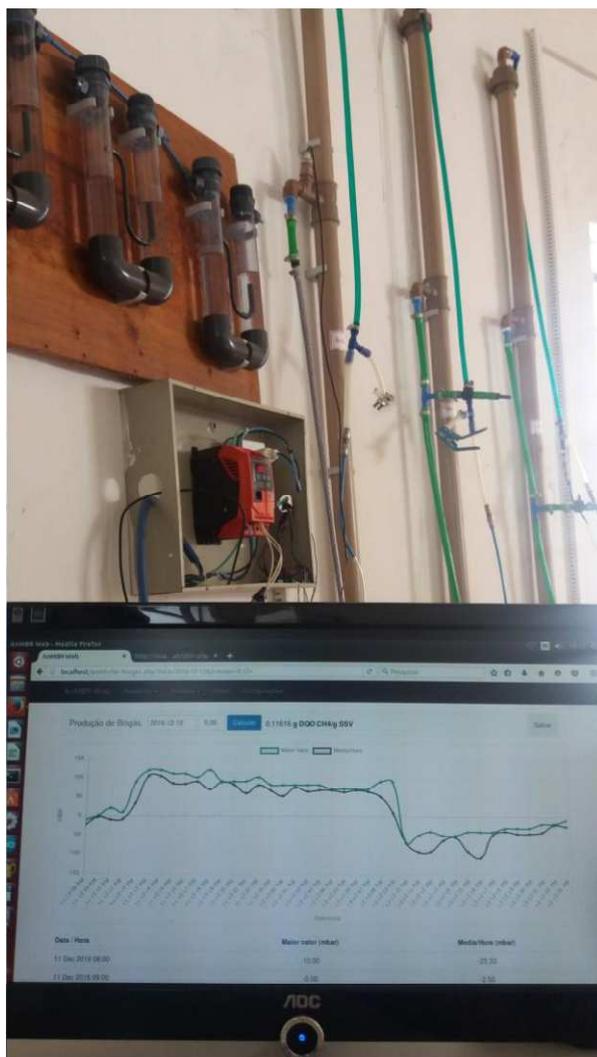
A pesquisa experimental foi realizada na Estação Experimental de Tratamentos Biológicos de Esgotos Sanitários – EXTRABES – UEPB, localizado no bairro do Tambor na cidade de Campina Grande – PB. A pesquisa foi realizada durante o período de um ano, teve início no mês fevereiro de 2016, e encerrou-se em fevereiro de 2017.

O sistema experimental foi executado em três etapas distintas. Na primeira etapa, foi realizado o processo de montagem inicial do biorreator de membrana, definindo os pontos específicos de instalação dos sensores, de pressão e temperatura no biodigestor, e sensores de pressão no módulo de membrana.

Na segunda etapa, foi realizada a instalação dos sensores no biorreator, os quais estão conectados a uma placa micro controladora Arduino conectada a um computador, permitindo o estabelecimento de uma conexão entre os dispositivos através de uma conexão USB.

A terceira etapa do projeto envolve o desenvolvimento do software, este permite ao operador do sistema acompanhar online e em tempo real as condições do biorreator. A figura 3 mostra o sistema em funcionamento.

Figura 3 - Sistema em funcionamento com resultados visualizados em tempo real.



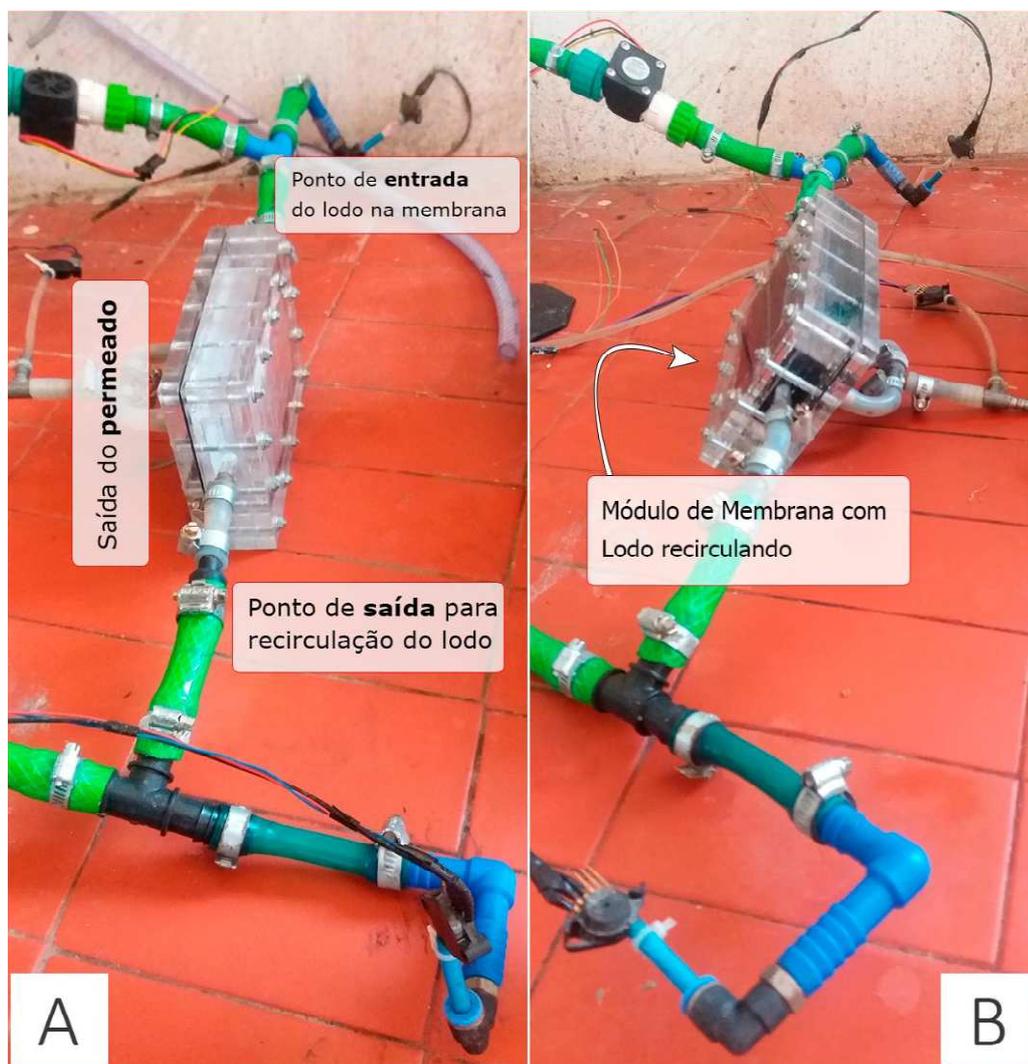
Fonte: Elaborado pelo autor.

4.2 Configurações do Biorreator de membrana

O sistema experimental inclui dois biorreatores, com características distintas de biomassa. Um biorreator foi operado com um módulo de membrana, neste módulo foi instalado três sensores de pressão da série MPX4250.

Na membrana, foi definido três pontos de instalação de sensores de pressão, o primeiro ponto é na entrada da membrana, o segundo na saída desta membrana e finalizando, no terceiro ponto tem-se um sensor de pressão na saída da membrana, onde é realizada a retirada do permeado. A seguir, é apresentado na figura 4 a membrana com os sensores instalados.

Figura 4 - (A) Membrana com sensores de pressão instalados, (B) Membrana com lodo em circulação



Fonte: Elaborado pelo autor.

A figura 4 (A) mostra o módulo de membrana vazio, antes de iniciar o processo de encher o biorreator com o lodo de esgoto, pode-se observar a cor transparente do módulo. Na figura 4 (B) o lodo já foi colocado dentro do biorreator e o módulo de membrana aparece com coloração escura, indicando que o lodo está passando pelo módulo.

Os sensores acoplados a membrana foram conectados a placa micro controladora Arduino através das portas analógicas de comunicação, cada sensor utiliza uma porta específica para conexão, cada placa Arduino possui seis portas disponíveis para conexões analógicas. Ainda nesta placa microncontroladora instalou-se um sensor medidor de temperatura, a fim de obter valores da temperatura do lodo no biorreator.

A placa Arduino é o elemento coordenador do sistema de biorreator de membrana, responsável pelo controle de todos os sensores eletrônicos utilizados no sistema sendo sua função receber e armazenar os dados emitidos pelos sensores.

Finalizando esta sessão de configuração, é exibido na Figura 5 o esquema físico utilizado na automação do sistema experimental.

4.3 Automação do Biorreator com a plataforma Arduino

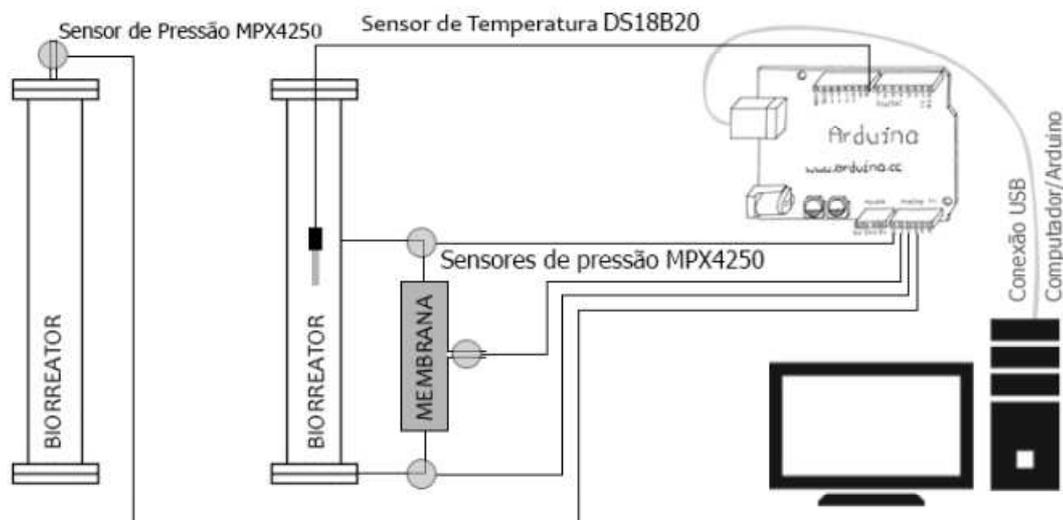
O intuito da automação do biorreator é realizar leituras automáticas dos parâmetros operacionais, os valores são obtidos há cada cinco minutos, onde em cada leitura são obtidos valores de pressão no módulo de membrana, temperatura do lodo no biorreator e pressão acumulada no biorreator conforme produção de biogás, possibilitando desta forma o armazenamento em um banco de dados.

Barbosa (2011), destaca a importância da utilização de um sistema de aquisição de dados que permita coletar amplas quantidades de dados emitidas por sensores para obter valores médios em larga escala e melhor precisão.

A inclusão da plataforma Arduino, em conjunto com alguns sensores, torna-se peça de grande importância na automação de sistemas de biorreatores, sendo esta placa responsável em fazer a comunicação entre os biorreatores e o computador. O esquema que envolve a conexão física entre a placa micro controladora Arduino, sensores, biorreator e módulo de membrana é apresentado na figura 5.

Conforme apresentado na figura 5, o sistema é composto por dois biorreatores. No primeiro foi instalado apenas um sensor de pressão para detectar a produção de biogás. O segundo reator é composto por um módulo de membrana, com três sensores de pressão, e um sensor de temperatura acoplado a este biorreator.

Figura 5 - Esquema de conexão física entre Biorreator, Módulo de Membrana, Computador servidor, Arduino e Sensores



Fonte: Elaborada pelo autor.

4.4 Conexão entre Arduino e sensores

Nesta placa micro controladora foram instalados três sensores da série MPX4250 no módulo de membrana e um no biorreator. Utilizando as portas analógicas zero, um e dois, respectivamente para instalação dos sensores no módulo de membrana na saída do permeado, entrada do lodo e saída do lodo para recirculação. Portanto após este procedimento de instalação, tem-se a placa Arduino conectada fisicamente aos sensores de pressão.

No biorreator, o sensor da série MPX4250 foi instalado com o objetivo de medir a pressão exercida sobre este sensor, conforme a produção de biogás no biorreator. No decorrer do dia a pressão no biorreator aumenta e é registrada progressivamente pelo sensor de pressão até atingir o valor limite de 2.0 bar, ao atingir este valor, o gás deve ser liberado através da via de purga.

Finalizando a instalação dos sensores, diferentemente dos sensores de pressão que utilizam a porta analógica da placa Arduino para conexão, o sensor de temperatura deve ser conectado a placa utilizando uma porta digital, neste projeto foi utilizado a porta digital 0 para conexão. No entanto, vale salientar que a conexão deste sensor de temperatura pode ser feita em qualquer uma das portas digitais disponíveis na placa.

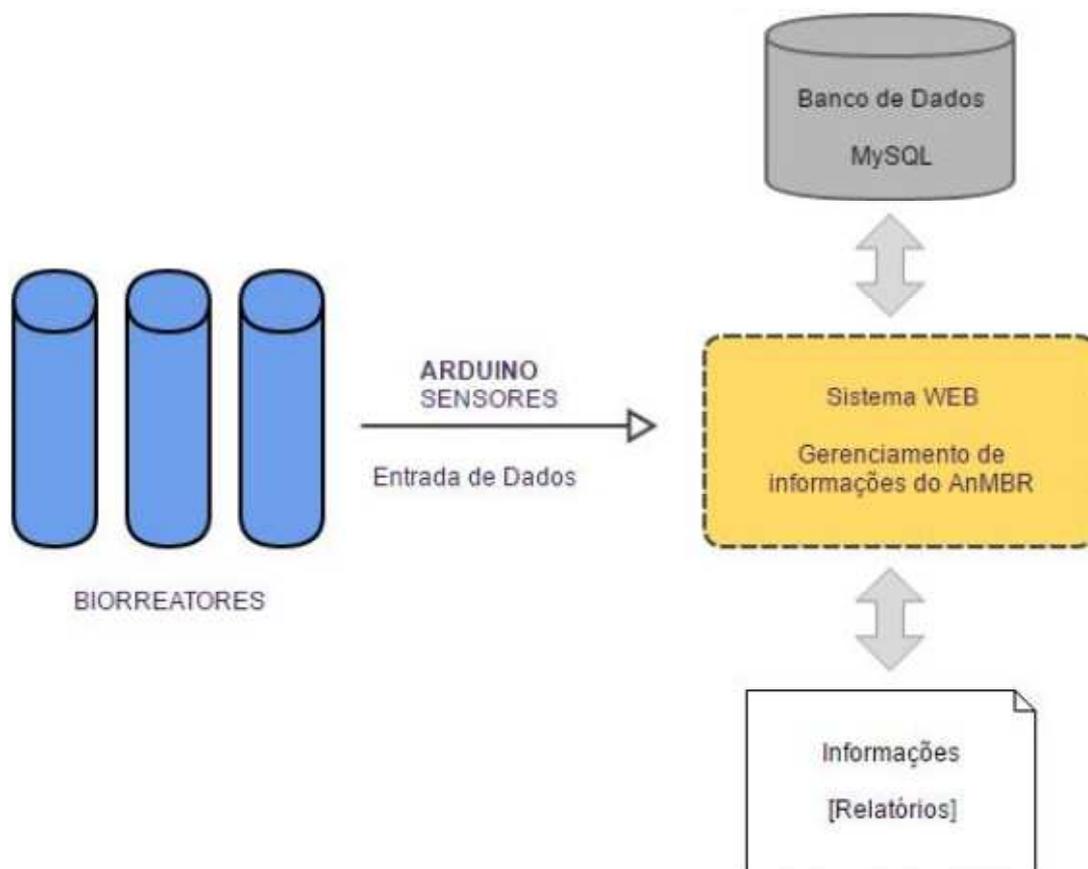
4.5 Estruturação do sistema de informação

A informação é constituída de vários dados relacionados e interpretados, sendo que o processamento desses possibilita a tomada de decisão para execução de uma tarefa específica, de acordo com os interesses do usuário final.

O sistema desenvolvido conta com gerência de informação sanitária ambiental, para auxílio na tomada de decisão, sendo que para o seu correto funcionamento existe a necessidade da alimentação dos dados, portanto para captação dos dados utilizou-se a placa micro controladora Arduino em conjunto com sensores de pressão e temperatura. A figura 6 ilustra o funcionamento entre as diferentes partes que envolvem o sistema, desde a captação dos dados através dos sensores conectados ao Arduino, até a fase de geração de relatórios, na interação do usuário com o software.

Inicialmente, os dados são captados pelos sensores que foram instalados nos biorreatores, a placa Arduino faz o controle dos dados recebendo as informações obtidas através dos sensores e armazenando-os em um banco de dados. Após armazenados os dados torna-se necessário uma ferramenta que permita o gerenciamento das informações, fazendo com que os valores obtidos e armazenadas sejam disponibilizadas em formato de relatórios.

Figura 6 - Conexão entre as partes que envolvem o sistema de informação.



Fonte: Elaborado pelo autor.

As informações são geradas com base no histórico dos dados registrados no banco de dados, auxiliando assim na composição de relatórios contendo informações para auxílio em tomada de decisões.

4.6 Desenvolvimento do Software Web

Nesta seção é apresentado configurações de hardware e softwares utilizados no desenvolvimento do projeto.

4.6.1 Equipamentos

O microcomputador responsável pelo armazenamento e processamento dos dados foi um microcomputador Celeron 2.4GHz, 1.5Gb de RAM, HD 500GB, sistema operacional Ubuntu Linux. Estas são as especificações

mínimas que podem ser utilizadas caso haja necessidade de montar um novo biorreator de membrana automatizado.

4.6.2 Linguagem computacional

As rotinas foram desenvolvidas de forma convencional, sempre priorizando a acessibilidade por usuários não especializados na área da informática, com acesso a diversos módulos do sistema através de menus autoexplicativos, tornando simples e fácil a operabilidade do sistema.

Os programas computacionais escolhidos para o desenvolvimento do projeto foram o Adobe Dreamweaver CS6 para a etapa de programação do aplicativo, com acesso ao banco de dados MySQL.

Para a criação da interface da aplicação foi usado HTML. Devido a necessidade de gerar gráficos e de se querer obter um nível avançado de interação e usabilidade, recorreu-se à especificação mais recente da linguagem, HTML5.

A linguagem de programação utilizada na implementação do software foi o PHP, com a justificativa das vantagens citadas no tópico 3.5.3, que traz uma fundamentação sobre esta linguagem.

4.6.3 Bando de Dados

Conforme características e vantagens citadas no item 3.5.4 deste trabalho, utilizou-se o MySQL como servidor de banco de dados. O banco de dados criado para o monitoramento do biorreator foi constituído de objetos (tabelas) relacionais.

Para construção das tabelas foi utilizado a linguagem SQL, o código para criação da estrutura do banco de dados, com linhas de comandos SQL de criação das tabelas do software, local de armazenamento dos dados do sistema é apresentado na figura 7.

Figura 7 - Código SQL com a estrutura de criação das tabelas do banco de dados

```
Query automacao.sql* x +
1
2 CREATE TABLE `usuario` (
3     `id` INT(11) NOT NULL,
4     `nome` VARCHAR(255) NOT NULL,
5     `email` VARCHAR(255) NOT NULL,
6     `senha` VARCHAR(255) NOT NULL,
7     `data_cadastro` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
8 ) ENGINE=MYISAM DEFAULT CHARSET=latin1;
9
10
11 CREATE TABLE `pressao_biogas` (
12     `id` INT(11) NOT NULL,
13     `data` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
14     `valor` FLOAT NOT NULL
15 ) ENGINE=INNODB DEFAULT CHARSET=latin1;
16
17
18 CREATE TABLE `pressao_membrana` (
19     `id` INT(11) NOT NULL,
20     `data` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
21     `p_saida` VARCHAR(20) NOT NULL,
22     `p_entrada` VARCHAR(20) NOT NULL,
23     `p_permeado` VARCHAR(20) NOT NULL
24 ) ENGINE=INNODB DEFAULT CHARSET=latin1;
25
26
27 CREATE TABLE `temperatura` (
28     `id` INT(11) NOT NULL,
29     `data` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
30     `valor` FLOAT NOT NULL
31 ) ENGINE=MYISAM DEFAULT CHARSET=latin1;
32
```

Fonte: Elaborado pelo autor.

Após executar o código SQL de criação das tabelas do sistema, o banco de dados está pronto para armazenar todas as informações que serão recebidas através dos sensores. Essas informações são armazenadas em tabelas, que após criadas seguem o formato apresentado na figura 8.

Figura 8 - Estrutura das tabelas do sistema

Tabela usuario				
id	nome	email	senha	data_cadastro
1	Thales Lacerda	thaleslqa@gmail.com	123456	2016-12-13 14:33:32
2	Fernanda Monte	fernandamonte20@gmail.com	123456	2016-12-13 14:35:10

Tabela pressao_membrana				
id	data	p_saida	p_entrada	p_permeado
1	2016-07-22 14:30:03	0.10	0.20	0.19
2	2016-07-22 14:35:03	0.10	0.20	0.18
3	2016-07-22 14:40:03	0.10	0.20	0.19
4	2016-07-22 14:45:03	0.11	0.20	0.13

Tabela temperatura		
id	data	valor
1	2016-12-13 14:46:14	26.88
2	2016-12-13 14:46:25	27.5
3	2016-12-13 14:46:36	28.3
4	2016-12-13 14:46:44	29
5	2016-12-13 14:46:54	30.1

Tabela pressao_biogas		
id	data	valor
2588	2016-11-14 11:11:03	0.11
2589	2016-11-14 11:12:03	0.09
2590	2016-11-14 11:13:03	0.1
2591	2016-11-14 11:14:03	0.1
2593	2016-11-14 11:16:03	0.09

Fonte: Elaborado pelo autor.

4.6.4 Frameworks

Um framework é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica, tem como principal objetivo tornar o desenvolvimento de um software mais ágil, trazendo e agregando novas funcionalidades ao projeto, evitando que o desenvolvedor perca tempo criando recursos que já existem prontos.

O projeto de software Web apresentado foi escrito utilizando os padrões do framework Bootstrap. Este torna o desenvolvimento de sites e aplicações Web mais rápido e fácil. Ele é feito para pessoas de todos os níveis e dispositivos de qualquer forma ou tamanho. Fácil e eficiente, Bootstrap adapta seus projetos Web com um simples código, de celulares para tablets e para desktops através de códigos pré-definidos.

Todas as configurações de layout do sistema, como estilo de fonte, cores de janela, cores de texto e links foram escritas adotando os padrões bootstrap,

com a proposta de apresentar um sistema com boa acessibilidade e ao mesmo tempo elegante.

Outro framework adotado no projeto, o Chart JS é um poderoso framework para gerar gráficos de diversos tipos, simples e flexível este é indicado para designers e desenvolvedores.

Chart JS é um framework de código aberto, projeto mantido pela comunidade, aberto a contribuições, possui oito diferentes maneiras de visualizar os dados, cada uma com animação e customização, apresenta grande desempenho de renderização em todos os navegadores modernos e redimensiona os gráficos na janela conforme características do dispositivo utilizado.

4.7 Estrutura do software AnMBR Web

O software AnMBR Web está organizado com a seguinte estrutura de diretórios e arquivos, conforme apresentado na figura 9.

Figura 9 - Estrutura de diretórios e arquivos do software AnMBR Web.

Nome	Data de modificaç...	Tipo	Tamanho
classes	09/01/2017 23:14	Pasta de arquivos	
crontab	21/11/2016 15:43	Pasta de arquivos	
css	25/12/2016 21:42	Pasta de arquivos	
images	21/11/2016 15:43	Pasta de arquivos	
js	21/11/2016 15:43	Pasta de arquivos	
config.php	25/12/2016 22:20	Arquivo PHP	2 KB
exporta-biogas.php	16/12/2016 12:23	Arquivo PHP	2 KB
exporta-ptm.php	19/09/2016 21:04	Arquivo PHP	2 KB
exporta-xls.php	30/03/2016 20:51	Arquivo PHP	2 KB
graficos.php	25/12/2016 18:02	Arquivo PHP	6 KB
header.php	23/12/2016 16:34	Arquivo PHP	3 KB
index.php	09/01/2017 23:20	Arquivo PHP	5 KB
ler-biogas.php	25/12/2016 22:13	Arquivo PHP	5 KB
ler-ptm.php	21/11/2016 23:37	Arquivo PHP	5 KB
ler-temperatura.php	09/01/2017 23:14	Arquivo PHP	5 KB
sobre.php	29/03/2016 18:18	Arquivo PHP	1 KB

Fonte: Elaborado pelo autor.

Diretório classes – Contém todos os arquivos de classes, onde tem-se a classe de conexão com o banco de dados, e as classes pressão de biogás e pressão da membrana, como também a classe temperatura. No apêndice B é apresentado o código de criação de todas as classes do sistema.

Diretório crontab – contém o arquivo que é executado conforme um determinado intervalo de tempo, este intervalo pode ser editado através do menu configurações no software AnMBR Web, por padrão é definido o tempo de cinco minuto, ou seja, a cada cinco minutos um novo valor é armazenado.

Diretório css – Aqui são incluídos todos os arquivos referentes as folhas de estilos criadas durante o desenvolvimento da aplicação. Como a opção de desenvolvimento foi adotada o framework bootstrap, torna-se necessário incluir o arquivo bootstrap.min.js para utilização deste framework.

Diretório js – Nesta pasta foi adicionado todos os arquivos com extensões (.js). Ao utilizar o framework bootstrap também é necessário incluir o arquivo bootstrap.min.js, portanto este arquivo é adicionado ao diretório js do projeto desenvolvido.

Após apresentação dos diretórios que fazem parte do sistema, tem-se a seguir uma descrição dos arquivos que estão na pasta raiz do sistema:

Arquivo index.php – Arquivo responsável por apresentar as informações na página inicial do sistema.

Arquivo header.php – Arquivo de cabeçalho, onde são incluídos arquivos externos necessários para a utilização dos frameworks utilizados.

Arquivo ler-biogás.php – Nesta página são apresentadas informações referente a produção de biogás no biorreator.

Arquivo ler-temperatura.php – Página onde são apresentadas informações referente a temperatura do lodo no biorreator.

Arquivo ler-ptm.php – Nesta página pode-se visualizar os valores referentes a pressão da membrana, são exibidos valores de pressão do permeado, pressão na entrada da membrana e pressão na saída da membrana para recirculação. Também é exibido nesta página o valor da pressão transmembrana, que é calculado da seguinte maneira: soma das pressões de entrada e saída, divide-se por dois, deste valor subtrai a pressão do permeado.

Arquivo `exporta-ptm.php` – Responsável por executar uma sequência de comandos que disponibiliza ao usuário a opção de exportar o relatório o qual está sendo visualizado para um arquivo de planilha.

Arquivo `graficos.php` – Gera gráficos da pressão exercida nos três pontos da membrana, onde o usuário tem a data atual pré-selecionada, porém é disponibilizado uma opção de filtrar por data, caso o usuário deseje selecionar uma data específica. Dois gráficos são apresentados nesta página, o primeiro faz uma média diária dos valores obtidos, e o segundo, traz uma média para cada 12 horas. Os valores são apresentados no gráfico que traz uma relação entre data e pressão (mBar).

Arquivo `sobre.php` – Página estática que exibe informações gerais do projeto, como título, autor e data de criação.

4.8 Metodologia desenvolvida para cálculo da produção de biogás

Há diferentes métodos para medição de biogás produzido em testes de atividade metanogênica específica, os quais podem ser classificados em manométricos ou volumétricos.

Segundo Aquino et al. (2007), a grande vantagem do método manométrico é a possibilidade de se acoplarem os medidores de pressão a microcomputadores, permitindo assim o monitoramento instantâneo e a automação do processo, enquanto que a principal desvantagem está relacionada ao custo de aquisição e manutenção dos equipamentos. No entanto pode-se destacar que, devido a utilização da plataforma Arduino, o qual é considerada uma alternativa de baixo custo, a desvantagem de custo elevado citada anteriormente pode ser desconsiderada.

Com o sistema automatizado, no decorrer do dia a pressão no biorreator aumenta, conforme a capacidade máxima de produção de metano por um consórcio de microrganismos anaeróbios, realizando a conversão de substratos orgânicos a biogás. Conforme a pressão aumenta, seu valor é registrado progressivamente pelo sensor de pressão até atingir o valor limite de 2.0 bar, ao atingir este valor, o gás deve ser liberado através da via de purga. Isso faz com que a pressão do sistema diminua devido à liberação do gás acumulado no biorreator.

O método desenvolvido se baseia na medição da pressão exercida sobre um sensor (da série MPX4250) acoplado ao biorreator. Dependendo da configuração do sistema a pressão medida pode ser devido à mistura de gases do biogás, que é constituído principalmente por metano e dióxido de carbono, ou devido somente ao metano. A figura 10 apresenta o sensor de pressão acoplado ao biorreator com a finalidade de medir a produção de biogás.

Figura 10 - Sensor de Pressão instalado no biorreator para medir a produção de Biogás



Fonte: Elaborado pelo autor.

Conforme Aquino et al. (2007), é importante ressaltar que os procedimentos manométricos só dispensam o uso de cromatógrafo caso haja a absorção do dióxido de carbono em etapa prévia à medição de pressão. Nos respirômetros utilizados por Chernicharo et al (1997), a pressão registrada correspondia às pressões parciais do gás carbônico e metano e, por isso a determinação

cromatográfica dos gases era necessária para quantificação do metano produzido.

5 RESULTADOS E DISCUSSÕES

Os resultados e discussões apresentados têm como principal objetivo demonstrar de forma simplificada as principais ferramentas do sistema computacional desenvolvido.

5.1 Descrição do sistema

AnMBR WEB é o nome do sistema operacional desenvolvido para monitoramento das informações de biorreatores de membrana, com o objetivo de auxiliar na tomada de decisões, através de relatórios gerados em tempo real. A figura 11 é apresentado o menu de acesso do software AnMBR, com as funções disponibilizadas pelo sistema.

Figura 11 - Menu de acesso do software AnMBR



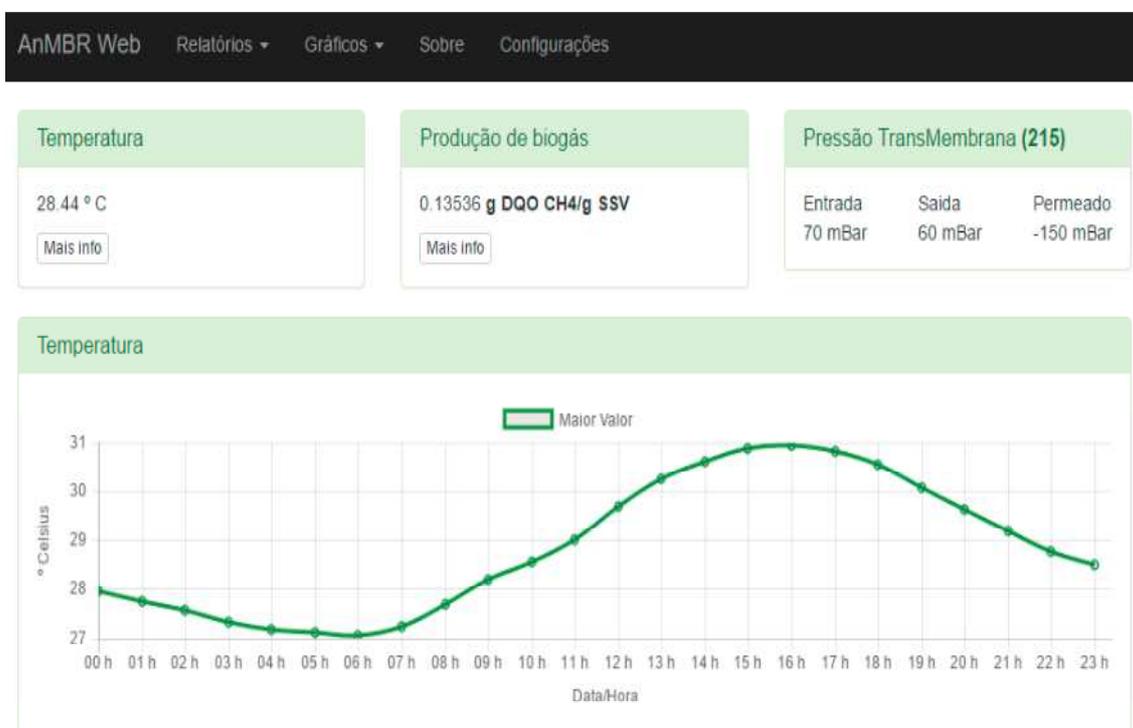
Fonte: Elaborada pelo autor.

No próximo tópico são apresentados os módulos que compõe o sistema, como também é apresentado todo o código fonte que se refere a criação de cada página do sistema.

5.2 Página inicial

Ao acessar o software, através do endereço [HTTP://localhost/anmbr](http://localhost/anmbr), é apresentado ao usuário na página inicial um resumo com as informações mais recentes obtidas através dos sensores, nesta tela tem-se valores obtidos em tempo real de temperatura, produção de biogás e pressão transmembrana. Os valores de temperatura são apresentados no formato de texto, como também em formato gráfico. A página inicial do software pode ser vista abaixo, conforme figura 12.

Figura 12 - Página inicial do software



Fonte: Elaborada pelo autor.

5.3 Página configurações

Nesta página, cabe ao usuário do sistema configurar valores que são utilizados para realizar o cálculo da produção de biogás. O operador do sistema deve preencher os seguintes campos: volume de lodo diário, concentração do lodo e head space. Esta página pode ser observada na figura 13.

Cabe ao operador do sistema, preencher as informações apresentadas no formulário de configurações de acordo com as especificações utilizadas no experimento. Após definir os valores no formulário, o usuário deve clicar no botão salvar para que as novas configurações sejam salvas.

Figura 13 - Página Configurações.

AnMBR Web	Relatórios ▾	Gráficos ▾	Sobre	Configurações
Configurações				
Volume de lodo diário	<input type="text" value="100"/>	mL		
Concentração do lodo	<input type="text" value="20"/>	g/L SSV		
HEAD Space	<input type="text" value="0.763"/>	Litros		
<input type="button" value="Salvar"/>				

Fonte: Elaborado pelo autor.

5.4 Módulo relatórios

Neste módulo, ao acessar o menu relatórios o usuário pode gerar três diferentes tipos, as seguintes opções são disponíveis: produção de biogás, temperatura e pressão transmembrana.

5.4.1 Produção de biogás

Nesta seção, os dados apresentados são formatados em uma tabela HTML, exibindo uma listagem com informações relacionadas a produção de biogás no reator, no primeiro acesso são listadas informações relacionadas ao dia atual, no entanto, o operador do sistema tem como opção selecionar uma data específica.

Para cada data selecionada, o software gera um novo gráfico, ao selecionar uma nova data, o gráfico é gerado com informações referente as últimas quarenta e oito horas, iniciando-se a partir da data escolhida.

Na parte superior desta página, é apresentado um gráfico que traz valores de data e hora em função da pressão, o valor da pressão é apresentado no formato milibar (mBar).

As leituras são realizadas a cada cinco minutos, para cada hora é realizada uma média dos valores lidos. Duas curvas dos valores gerados são apresentadas no gráfico, uma representa o maior valor lido em cada hora, e a outra representa uma curva com a média dos valores. Um gráfico de produção de biogás é apresentado a seguir na figura 14. Abaixo do gráfico são apresentados os mesmos valores em formato de tabela.

Conforme ilustrado no gráfico apresentado na figura 14, ao iniciar um novo ciclo de medição de biogás, a pressão aparece com valores próximo a zero. Ao decorrer do dia, após oito horas de experimento, a pressão no biorreator aumenta, chegando até 80 bar, esse pico na pressão pode ser observado no gráfico. Após este pico, o valor da pressão se estabiliza na faixa de 50 bar e se mantém durante as próximas doze horas. Portanto, pode-se destacar que neste momento em que a pressão se mantém estabilizada pode ser importante para que o operador do sistema possa perceber qual será a taxa máxima de produção de biogás.

O gráfico apresentado na figura 14 traz registros diário da pressão o que permite determinar a taxa diária de produção de metano. A produção volumétrica de metano é calculada diariamente, multiplicando-se o volume de biogás pela porcentagem de metano no biogás.

Figura 14 - Módulo Relatório Produção de Biogás.



Fonte: Elaborado pelo autor.

5.4.2 Pressão transmembrana

Módulo de relatório responsável por listar os valores nos três diferentes pontos da membrana, os valores são obtidos a cada cinco minutos e são apresentados em formato de lista. Para cada leitura realizada tem-se os valores de pressão nos três pontos da membrana, entrada, saída e permeado.

Este relatório é de grande importância no monitoramento de biorreatores de membrana, pois através das pressões apresentadas neste relatório, cabe ao operador do sistema tomar uma decisão na medida que os valores obtidos estão variando ou fora do valor desejado. Quando o valor da pressão aumenta muito, atingindo valores acima de 1000 bar, o operador do sistema deve ficar em alerta e decidir qual decisão tomar para normalizar a pressão na membrana.

A pressão transmembrana (PTM) é a diferença de pressão entre o lado da alimentação da membrana e o lado do permeado (MAESTRI, 2007, p. 45). O valor da pressão transmembrana apresentado neste relatório é calculado utilizando a seguinte equação (1):

$$PTM = (P_{\text{entrada}} + P_{\text{saída}})/2 - P_{\text{permeado}} \quad (1)$$

Conforme figura 15, que apresenta um relatório de pressão transmembrana, na primeira coluna tem-se a data e hora, na segunda, terceira e quarta coluna, são listadas pressões da entrada, saída e permeado, respectivamente.

O usuário tem a opção de selecionar a data desejada para exibir o relatório, como também é disponibilizada uma opção onde é possível salvar os dados em um arquivo de planilha, apenas clicando no botão Salvar.

AnMBR Web Relatórios ▾ Gráficos ▾ Sobre Configurações

Pressão Transmembrana 2016-11-01 Salvar

Data / Hora	Entrada (mbar)	Saída (mbar)	Permeado (mbar)	Transmembrana (mbar)
1 Nov 2016 12:00 AM	190	140	80	85
1 Nov 2016 12:05 AM	190	140	80	85
1 Nov 2016 12:10 AM	180	140	80	80
1 Nov 2016 12:15 AM	190	150	80	90
1 Nov 2016 12:20 AM	180	150	90	75
1 Nov 2016 12:25 AM	180	140	80	80
1 Nov 2016 12:30 AM	180	140	80	80
1 Nov 2016 12:35 AM	190	140	80	85
1 Nov 2016 12:40 AM	180	150	100	65
1 Nov 2016 12:45 AM	170	140	80	75
1 Nov 2016 12:50 AM	190	150	80	90
1 Nov 2016 12:55 AM	180	150	80	85

Figura 15 - Módulo relatório Pressão Transmembrana.

5.4.3 Temperatura

Na escolha deste tipo de relatório, é apresentado uma listagem com os últimos valores de temperatura no biorreator, podendo o usuário filtrar seus dados de acordo com a data desejada. Ainda nesta página, o usuário tem a opção de exportar os dados apresentados para um arquivo de planilha.

Conforme ilustrado na figura 16, na página relatório de temperatura, são apresentados valores da temperatura durante o dia, na parte superior desta página é plotado um gráfico com os valores obtidos pelo sensor de temperatura. O sistema faz uma média dos valores obtidos para cada hora e também exibe o maior e o menor valor adquirido pelo sensor.

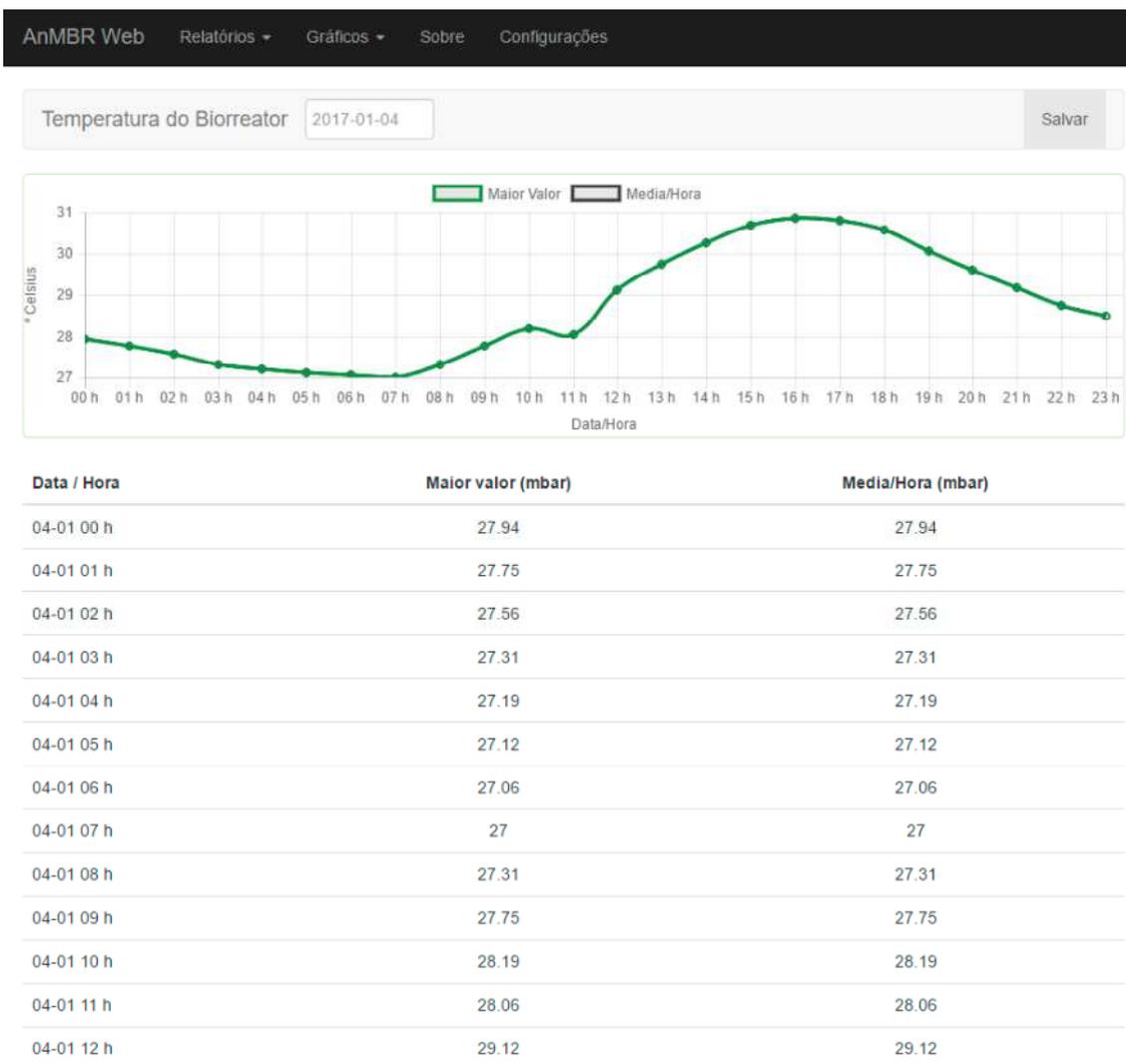


Figura 16 - Relatório Temperatura

5.4.4 Gráfico Pressão Transmembrana

Ao acessar o menu, clicando na opção gráficos, o usuário tem disponível uma página onde são exibidos valores em formato gráficos relacionados aos valores de pressão na membrana. Também neste modulo tem-se a opção de filtrar os dados de acordo com uma data desejada, como também a opção para salvar os dados em arquivo no formato de planilha.

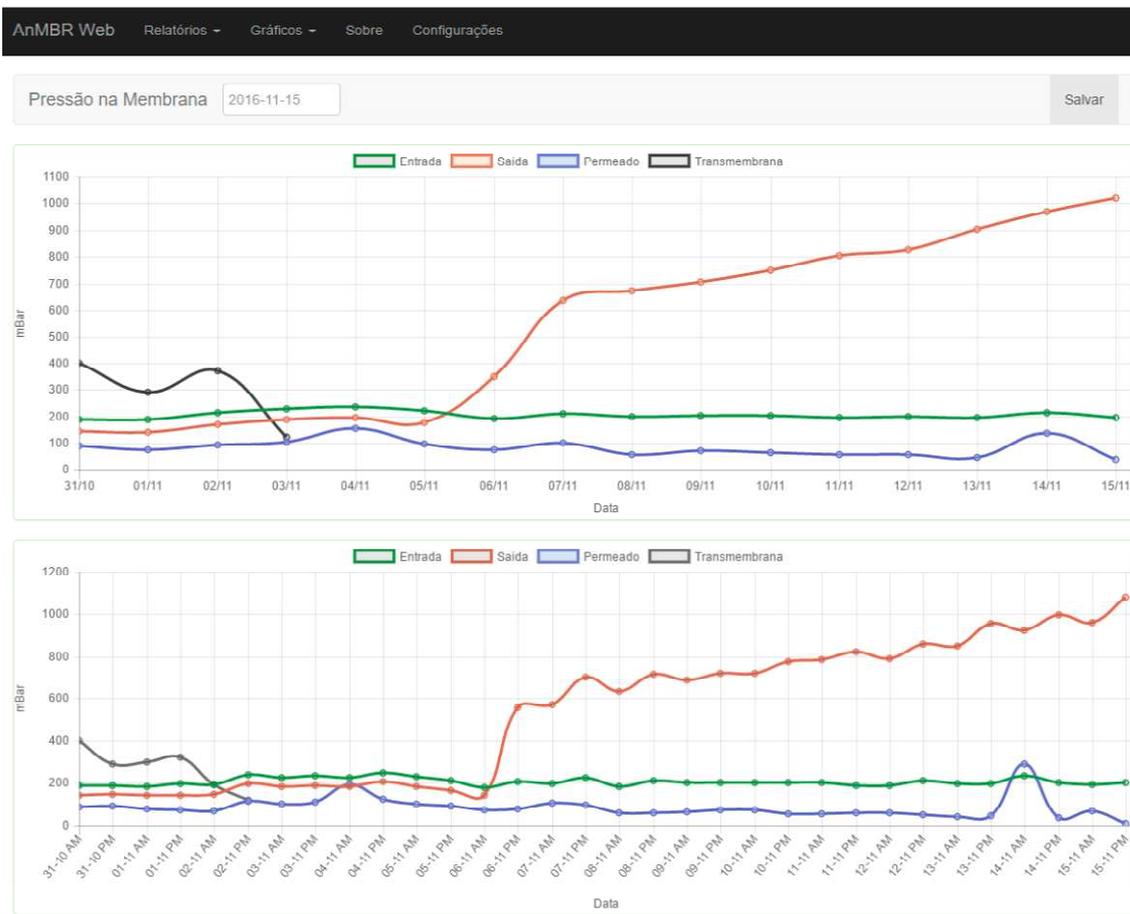


Figura 17 - Gráfico Pressão Transmembrana

Nesta página dois gráficos são apresentados, pode-se observar na figura 17. Os gráficos trazem valores da data em função da pressão exercida no ponto específico da membrana. A diferença entre eles é que no primeiro gráfico os valores são exibidos com intervalo de 24 horas, o segundo traz o intervalo de 12 horas.

Conforme ilustrado na figura 17, observe que a pressão na saída da membrana aumenta progressivamente durante um intervalo de 10 dias. Este aumento ocorreu porque a membrana chegou em um processo conhecido como incrustação.

A Incrustação é um processo físico que acontece devido à formação de uma camada de partículas (torta) sobre a membrana, provocando o decaimento do fluxo de permeação ao longo do tempo de operação (PELEGRIN, 2004). Esse fenômeno também é conhecido como fouling. Segundo Radjenovic et al. (2008) se trata de um fenômeno muito complexo, com diversos fatores intervenientes, além disso, é difícil defini-lo claramente. Esses autores citam como principal causa da incrustação os seguintes fatores: adsorção de macromoléculas e material coloidal; crescimento de biofilme na superfície da membrana e precipitação de matéria orgânica ou envelhecimento da membrana.

A identificação da incrustação da membrana é de fundamental importância para um desempenho adequado do biorreator. É através desta identificação que se pode estimar o momento da limpeza da membrana. Sistemas com operação com pressões elevadas na saída da membrana podem gerar um maior consumo de energia, uma colmatação mais rápida ou irreversível podendo até causar danos à membrana.

Portanto, pode-se destacar que o software desenvolvido neste projeto apresenta uma característica importante neste processo, mostrando ao operador do sistema qual o momento adequado para realizar uma limpeza na membrana.

6 CONCLUSÕES

O objetivo principal proposto no presente trabalho foi atendido com automação de biorreatores utilizando a placa micro controladora Arduino, utilizando um sensor de temperatura e pressão acoplado ao reator, como também três sensores de pressão instalados no módulo de membrana.

O início da pesquisa, envolveu a parte de montagem dos biorreatores, instalação dos sensores, da placa Arduino e do computador que armazena as informações referentes aos biorreatores.

Na segunda fase da pesquisa foi realizado o desenvolvimento do software, que define como as informações obtidas pelos sensores serão apresentadas em uma tela de computador. O software desenvolvido, permite o gerenciamento de biorreatores de membrana e apoio a decisão que reuniu, em um contexto, acesso a dados e informações de forma simples e eficiente, agregando alta disponibilidade e acessibilidade.

Os resultados obtidos apontam a importância da informatização de diversas áreas da engenharia ambiental de forma consciente e planejada.

Diante das possibilidades do gerenciamento de informações e auxílio à decisão na gestão, destaca-se a elaboração de relatórios em tempo real e gráficos com consistência e confiabilidade das informações, oferecendo ainda nível aceitável de segurança dos dados e informações.

Ao término da pesquisa foi possível constatar que a automação do sistema facilita a armazenagem e tratamento de dados, oferecendo um melhor gerenciamento das informações de extrema importância no auxílio à tomada de decisão na gestão de biorreatores de membrana.

REFERÊNCIAS

AQUINO, S. F.; CHERNICHARO, C. A.; FORESTI, E.; DOS SANTOS, M. L. F.; MONTEGGIA, L. O. Metodologias para determinação da atividade metanogênica específica (AME) em lodos anaeróbios. *Engenharia Sanitária Ambiental*, v. 12, n. 2, p. 192-201, 2007.

BANZI, M.; SHILOH, M. Primeiros passos com o Arduino. 2ª Edição. São Paulo, SP. Novatec Editora, 2010.

BARBOSA, M. A. G. Calibração e uso de sensores FDR para determinação da umidade e salinidade em dois tipos de solos. Dissertação (Mestrado). Programa de Pós-Graduação Irrigação e Drenagem, Universidade Federal Rural do Semiárido, Mossoró – RN, 2011.

BELLI, T. J.; AMARAL, P. A. P.; RECIO, M. A. L.; VIDAL, C. M. S.; LAPOLLI, F. R. Biorreator à membrana em batelada sequencial aplicado ao tratamento de esgoto visando à remoção de nutrientes. *Engenharia Sanitária Ambiental, Florianópolis - SC.*, v. 17, n. 2, 2012. p. 143-154.

CHERNICHARO, C. A. L. Reatores Anaeró-bios - Princípios do tratamento biológico de águas residuárias, v. 5, DESA-UFMG, 1997.

DE SOUSA, C. A. Tratamento termofílico aeróbio de efluente de máquina de papel utilizando biorreator a membranas. Dissertação (Doutorado). Programa de Pós-Graduação em Ciência Florestal, Universidade Federal de Viçosa, Viçosa – MG, 2008.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. Arduino em ação. 1ª Edição. São Paulo – SP. Novatec Editora, 2013.

Freescale Semiconductor Inc. Integrated silicon pressure sensor on-chip signal conditioned, temperature compensated and calibrated: MPX4250 Series, 7th edn, Tempe, AZ, 2009. Disponível em <http://www.nxp.com/assets/documents/data/en/data-sheets/MPX4250.pdf>.

GALVÃO, W. C. Sistema web para gerenciamento de informações agrícolas. Dissertação (Mestrado). Universidade Estadual Paulista, Faculdade de Ciências Agrônômicas. Botucatu - SP, 2010.

GARCIA, J. G. M. Desenvolvimento de uma plataforma web para aplicações de cálculo estrutural. Dissertação (Mestrado). Faculdade de Engenharia da Universidade do Porto. Portugal, 2012.

GERARDO, M. L.; ZACHAROF, M. P.; LOVITT, R. W. Strategies for the recovery of nutrients and metals from anaerobically digested dairy farm sludge using cross-flow microfiltration. *Water Research*, n. 47, p. 4833-4842, 2013.

GHANI, M. F.; ABDULLAH, S. S. Design of a Body with Depth Control System for an Underwater Glider. *Journal of Ocean, Mechanical and Aerospace. Science and Engineering*, v. 3, p. 20, 2014.

FABRI JUNIOR, L. A. O uso de Arduino na criação de kit para oficinas de robótica de baixo custo para escolas públicas. Faculdade de Tecnologia, Universidade Estadual de Campinas, Limeira – SP, 2014.

LIU, Z. J. Multi Point Temperature Measurement System Based on DS18B20. *Advanced Materials Research*, v. 756-759, p. 556-559, 2013.

MAESTRI, R. S. Biorreator à Membrana como alternativa para o tratamento de esgotos sanitários e reuso de água. Programa de Pós-Graduação em Engenharia Ambiental, Universidade Federal de Santa Catarina, 2007.

MCROBERTS, M. *Arduino Básico*. 1ª Edição. São Paulo – SP. Novatec Editora, 2011.

MIERZWA, J. C. et al. Tratamento de água para abastecimento público por ultrafiltração: avaliação comparativa através dos custos diretos de implantação e operação com os sistemas convencional e convencional com carvão ativado. *Engenharia Sanitária Ambiental*, v. 13, n. 1, p. 78-87, 2008.

MONK, S. *Programação com Arduino: começando com sketches*. 1ª edição. Porto Alegre – RS. Bookman, 2013.

PELEGRIN, D. C. Microfiltração tangencial de efluente sanitário após tratamento biológico. Dissertação (Mestrado). Programa de Pós-graduação em Engenharia Ambiental, Universidade Federal de Santa Catarina, Florianópolis, 2004.

SOARES, W. *PHP5: Conceitos, Programação e Integração com Banco de Dados*. 3ª edição. São Paulo – Sp. Editora Érica Ltda, 2007.

PROVENZI, G. Biorreator à membrana submersa para o tratamento biológico de efluentes: Estudos hidrodinâmicos e físico-químicos no controle de colmatação. Programa de Pós-Graduação em Engenharia Ambiental Universidade Federal de Santa Catarina Florianópolis – SC, 2005.

RADJENOVIC, J. et al. Membrane bioreactor (MBR) as an advanced wastewater treatment technology. In: *Handbook Environmental Chemistry*. Springer-Verlag Berlin Heidelberg, v. 5, 2008. p.37-101.

RAMALHO, J. A. A. *Iniciando em HTML*. São Paulo - SP: 1996.

SCHNEIDER, R. P.; TSUTIYA, M. T. Membranas filtrantes para o tratamento de água, esgoto e água de reuso. ABES, 2001.

VIANA, P. Z. Biorreator com membrana aplicado ao tratamento de esgotos domésticos: avaliação de desempenho de módulos de membranas com circulação externa. Programa de Pós-Graduação em Engenharias, Universidade Federal do Rio de Janeiro Rio de Janeiro – RJ, 2004.

APÊNDICES

APÊNDICE A – Página inicial do servidor Web

```

1 <?php
2 require_once("header.php");
3 require_once("classes/classes.php");
4
5 $classObj = new Comando();
6 $objTemp = new Temperatura();
7 ?>
8 <div class="container">
9   <div class="row">
10     <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
11       <div class="panel panel-success">
12         <div class="panel-heading">
13           <h3 class="panel-title">Temperatura</h3>
14         </div>
15         <div class="panel-body">
16           <p><?php echo $objTemp->getTemperatura() ?> ° C</p>
17           <a class="btn btn-default btn-xs" href="#" role="button">Mais info</a>
18         </div>
19       </div>
20     </div>
21     <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
22       <div class="panel panel-success">
23         <div class="panel-heading">
24           <h3 class="panel-title">pH</h3>
25         </div>
26         <div class="panel-body">
27           <p>7.0</p>
28           <a class="btn btn-default btn-xs" href="#" role="button">Mais info</a>
29         </div>
30       </div>
31     </div>
32     <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
33       <div class="panel panel-success">
34         <div class="panel-heading">
35           <h3 class="panel-title">Pressão transmembrana</h3>
36         </div>
37         <div class="panel-body">
38           <div class="row">
39             <div class="col-xs-4 col-sm-4 col-md-4 col-lg-4">PTM 1</div>
40             <div class="col-xs-4 col-sm-4 col-md-4 col-lg-4">PTM 2</div>
41             <div class="col-xs-4 col-sm-4 col-md-4 col-lg-4">PTM 3</div>
42           </div>
43           <div class="row">
44             <div class="col-xs-4 col-sm-4 col-md-4 col-lg-4">0 bar</div>
45             <div class="col-xs-4 col-sm-4 col-md-4 col-lg-4">0 bar</div>
46             <div class="col-xs-4 col-sm-4 col-md-4 col-lg-4">0 bar</div>
47           </div>
48         </div>
49       </div>
50     </div>
51 </div><!-- /row -->
52
53 <div class="panel panel-success">
54   <div class="panel-heading">
55     <h3 class="panel-title">Leitura de BioGas</h3>
56   </div>
57   <div class="panel-body">
58     <table class="table table-hover">
59       <thead>
60         <tr>
61           <th>Data / Hora</th>
62           <th>Medidor</th>
63           <th class="text-center">Temperatura</th>
64           <th class="text-center">Fluxo [mL/d]</th>
65           <th class="text-center">Volume [mL]</th>
66           <th class="text-center">Pressão (mbar)</th>
67         </tr>
68       </thead>
69       <tbody>
70         <?php
71         $sqlListar = "SELECT * FROM `leitura-gas` ORDER BY ID DESC LIMIT 6";
72         $result = $classObj->Executar( $sqlListar);
73         while($dados = $result->fetch_array()){
74           ?>
75           <tr>
76             <td><?php echo date('j M Y g:i A', strtotime($dados['time_start'])); ?></td>
77             <td><?php echo $dados['id_medidor'] ?></td>
78             <td class="text-center"><?php echo $dados['temperature'] ?></td>
79             <td class="text-center"><?php echo $dados['gas_flow'] ?></td>
80             <td class="text-center"><?php echo number_format($dados['gas_acumulation'], 4) ?></td>
81             <td class="text-center"><?php echo $dados['atm_mbar'] ?></td>
82           </tr>
83         </tbody>
84       </table>
85     </div>
86   </div>
87 </div>
88 </div><!-- /container -->
89
90 </body>
91 </html>
92

```

APÊNDICE B – Arquivo de importação com todas as classes utilizadas

```
1 <?php
2
3 require_once("Conexao.php");
4 require_once("Comando.php");
5 require_once("Pressao.php");
6 require_once("PressaoBiogas.php");
7 require_once("Temperatura.php");
8
9 ?>
10
```

APÊNDICE C – Classe Conexao.php

```
1 <?php
2
3 class Conexao {
4
5     var $host    = "localhost";
6     var $usuario = "root";
7     var $senha   = "";
8     var $banco   = "automacao";
9
10    public $mysqli;
11
12    public function Abrir() {
13        $this->mysqli = new mysqli($this->host, $this->usuario, $this->senha, $this->banco);
14    }
15
16    public function Fechar() {
17        $this->mysqli->close();
18    }
19 }
20
21 ?>
```

APÊNDICE D - Classe Comando.php

```
1 <?php
2
3 class Comando {
4     public function Executar($sql) {
5         $con = new Conexao();
6         $con->Abrir();
7         $re = $con->mysqli->query($sql);
8         $con->Fechar();
9         return $re;
10    }
11 }
12
13 ?>
```

APÊNDICE E – Classe Pressao.php

```

1  <?php
2  class Pressao {
3
4      public function getPressao($col){
5          $classObj = new Comando();
6          $result = $classObj->Executar("SELECT (".$col.")*1000 FROM 'pressao_membrana' ORDER BY fd DESC LIMIT 1");
7          $temp = $result->fetch_row();
8          return $temp[0];
9      }
10
11     public function dataLeituraPressao(){
12         $classObj = new Comando();
13         $result = $classObj->Executar("SELECT data FROM 'pressao_membrana' ORDER BY fd DESC LIMIT 1");
14         $temp = $result->fetch_row();
15         return $temp[0];
16     }
17
18     public function geraDados24H($seixo, $p_pressao, $data){
19
20         $classObj = new Comando();
21         $retorno = "";
22         $dataPadrao = new DateTime($data);
23         $dataInicio = $dataPadrao->modify('-15 day');
24         $dataF = new DateTime($data);
25         $dataFim = $dataF->modify('+1 day');
26         $sql = "SELECT * FROM (
27             SELECT data, DATE_FORMAT(data, '%d/%m') as data2,
28                 ROUND((sum(".$p_pressao."/)/COUNT(fd))*1000,2) as media_mBar
29             FROM pressao_membrana
30             WHERE
31                 data BETWEEN '".$dataInicio->format('Y-n-d')."' AND '".$dataFim->format('Y-n-d')."'
32             GROUP BY DATE_FORMAT(data, '%d-%m-%y')
33         ) AS tmp_table ORDER BY data ASC
34         ";
35         $result = $classObj->Executar($sql);
36         while($dados = $result->fetch_array()){
37             if($seixo == 'x') {
38                 $retorno .= "" . trim($dados['data2']) . ", ";
39             }elseif($seixo == 'y') {
40                 $retorno .= "" . $dados['media_mBar'] . ", ";
41             }
42         }
43         return $retorno;
44     }
45
46     public function geraDados12H($seixo, $p_pressao, $data){
47         $classObj = new Comando();
48         $retorno = '';
49         $dataPadrao = new DateTime($data);
50         $dataInicio = $dataPadrao->modify('-15 day');
51         $dataF = new DateTime($data);
52         $dataFim = $dataF->modify('+1 day');
53         $sql = "SELECT * FROM (
54             SELECT data, DATE_FORMAT(data, '%d-%m-%p') as data2, ROUND((SUM(".$p_pressao."/)/COUNT(fd)*1000),2) as media_mBar
55             FROM pressao_membrana
56             WHERE
57                 data BETWEEN '".$dataInicio->format('Y-n-d')."' AND '".$dataFim->format('Y-n-d')."'
58             GROUP BY DATE_FORMAT(data, '%d-%m-%y/%p')
59         ) AS tmp_table ORDER BY data ASC
60         ";
61         $result = $classObj->Executar($sql);
62         while($dados = $result->fetch_array()){
63             if($seixo == 'x') {
64                 $retorno .= "" . trim($dados['data2']) . ", ";
65             }elseif($seixo == 'y') {
66                 $retorno .= "" . trim($dados['media_mBar']) . ", ";
67             }
68         }
69         return $retorno;
70     }
71
72     public function geraPTH($intervalo){
73         $classObj = new Comando();
74         $retorno = '';
75         if($intervalo == '12H') {
76             $groupBy = "GROUP BY DATE_FORMAT(data, '%d-%m-%y/%p')";
77         } elseif($intervalo == '24H') {
78             $groupBy = "GROUP BY DATE_FORMAT(data, '%d-%m-%y')";
79         }
80         $sql = "SELECT * FROM (
81             SELECT data, DATE_FORMAT(data, '%d/%m') as data2,
82                 ROUND((SUM(p_entrada)/COUNT(fd) + SUM(p_saida)/COUNT(fd)/2) - p_perneado, 2)*1000 as trans_mBar
83             FROM pressao_membrana
84             WHERE
85                 data BETWEEN TIMESTAMP(DATE_SUB(NOW(), INTERVAL 15 day)) AND NOW() ".$groupBy."
86         ) AS tmp_table ORDER BY data ASC
87         ";
88         $result = $classObj->Executar($sql);
89         while($dados = $result->fetch_array()){
90             $retorno .= "" . $dados['trans_mBar'] . ", ";
91         }
92         return $retorno;
93     }
94 }

```

APÊNDICE F – Classe PressaoBiogas.php

```

1 <?php
2
3 class PressaoBiogas {
4
5     public function geraDados24H($coluna, $data){
6         $classObj = new Comando();
7         $retorno = "";
8         $sql = " SELECT DATE_FORMAT(data, '%d-%m %H %p') as data_out, ROUND(MAX(valor)*1000, 2) as maior_valor,
9                 ROUND((sum(valor)/COUNT(id))*1000,2) as media_pHora_mbar
10                FROM `pressao_biogas`
11                WHERE data BETWEEN DATE_ADD('".$data."', INTERVAL -16 HOUR) AND '".$data.'" 23:59:59'
12                GROUP BY DATE_FORMAT(data, '%d-%m-%y %H')
13                ";
14         $result = $classObj->Executar($sql);
15         while($dados = $result->fetch_array()){
16             if($coluna == 'data') {
17                 $retorno .= "" . trim($dados['data_out']) . ", ";
18             } elseif($coluna == 'maior_valor') {
19                 $retorno .= "" . $dados['maior_valor'] . ", ";
20             } elseif($coluna == 'media') {
21                 $retorno .= "" . $dados['media_pHora_mbar'] . ", ";
22             }
23         }
24         return $retorno;
25     }
26
27     public function calculaProducao($pressao){
28         $constR = 0.082;
29         $temp = 308;
30         $volHeadSpace = $this->getHeadSpace();
31         $volLodoDia = $this->getVolLodo();
32         $conclodo = $this->getConcentracaoLodo();
33         $n = number_format($pressao * $volHeadSpace/($constR * $temp), 5);
34         $massaMol = $n * 16;
35         $x_gDQO = (64 * $massaMol)/16;
36         $x_gSSV = ($volLodoDia * $conclodo)/1000;
37         $total = $x_gDQO/$x_gSSV;
38         echo $total . " <strong>g DQO CH4/g SSV</strong>";
39     }
40
41     public function getVolLodo() {
42         $classObj = new Comando();
43         $result = $classObj->Executar("SELECT vol_lodo FROM config;");
44         $temp = $result->fetch_row();
45         return $temp[0];
46     }
47
48     public function getConcentracaoLodo() {
49         $classObj = new Comando();
50         $result = $classObj->Executar("SELECT con_lodo FROM config;");
51         $temp = $result->fetch_row();
52         return $temp[0];
53     }
54
55     public function getHeadSpace() {
56         $classObj = new Comando();
57         $result = $classObj->Executar("SELECT head_space FROM config;");
58         $temp = $result->fetch_row();
59         return $temp[0];
60     }
61
62     public function getMaiorPressao($data) {
63         $classObj = new Comando();
64         $result = $classObj->Executar("SELECT MAX(valor) FROM pressao_biogas WHERE DATA LIKE '".$data."'");
65         $temp = $result->fetch_row();
66         return number_format($temp[0], 2);
67     }
68 }

```

APÊNDICE G – Classe Temperatura.php

```
1 <?php
2
3 class Temperatura {
4
5     public function getTemperatura(){
6         $classObj = new Comando();
7         $result = $classObj->Executar("SELECT valor FROM temperatura ORDER BY id DESC LIMIT 1;");
8         $temp = $result->fetch_row();
9         return $temp[0];
10    }
11
12    public function geraDados24H($coluna, $data) {
13        $classObj = new Comando();
14        $retorno = "";
15        $sql = " SELECT DATE_FORMAT(data, '%H h') as data, ROUND(MAX(valor), 2) as maior_valor,
16                ROUND((sum(valor)/COUNT(id)),2) as media_temp
17                FROM `temperatura`
18                WHERE data LIKE '". $data ."'
19                GROUP BY DATE_FORMAT(data, '%d-%m-%y %H') ";
20        $result = $classObj->Executar($sql);
21        while($dados = $result->fetch_array()){
22            if($coluna == 'data') {
23                $retorno .= "" . $dados['data'] . ", ";
24            } elseif($coluna == 'maior_valor') {
25                $retorno .= "" . $dados['maior_valor'] . ", ";
26            } elseif($coluna == 'media_temp') {
27                $retorno .= "" . $dados['media_temp'] . ", ";
28            }
29        }
30        return $retorno;
31    }
32 }
33
34 ?>
```