

Um Sistema Ubíquo para o Monitoramento Automático do Consumo de Água no Instituto Federal da Bahia, Campus Salvador

Patricia Miranda do N.Fonseca^{*}
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, S/N
Barbalho, Salvador - Bahia
patricia.fonseca@ieee.org

Manoel C. M. Neto[†]
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, S/N
Barbalho, Salvador - Bahia
manoelnetm@ifba.edu.br

Resumo

Neste projeto é apresentada a modelagem e desenvolvimento de um sistema de monitoramento de consumo de água aplicando fundamentais conceitos da Computação Ubíqua. Com uso de um sensor de vazão acoplado ao Arduino, plataforma de *hardware* com microcontrolador, dados de vazão são coletados e enviados ao servidor de banco de dados, onde são armazenados. Tais dados podem ser posteriormente visualizados através do acesso à aplicação *web* Portal SMA (Portal de Sustentabilidade e Meio Ambiente). O objetivo do sistema é monitorar o consumo de água no Instituto Federal da Bahia *campus* Salvador, a fim de criar políticas de economia dos recursos hídricos e consequentemente financeiros da instituição, além da conscientização do uso da água sem desperdícios.

Palavras-Chave

Computação ubíqua, Sistema de monitoramento, Arduino, PIMS

1. INTRODUÇÃO

Um dos principais problemas no Brasil é o desperdício de água, particularmente o que ocorre durante o processo de tratamento e transmissão de tal recurso aos consumidores brasileiros. Segundo dados do IBGE publicados em 2013, o Brasil perde cerca de 37% a 42% do total de sua água tratada e distribuída, principalmente nas grandes cidades, devido a vazamentos na rede de distribuição [22]. Esta realidade não é diferente no IFBA (Instituto Federal de Educação, Ciência e Tecnologia da Bahia), que por ter uma estrutura antiga (prédios com mais de 100 anos) contribui significativamente com esta perda.

A escassez da água é um problema grave que causa impactos ambientais importantes. Se não houver mudança no padrão de consumo, a população global poderá sofrer com a escassez de água até 2025, previsão declarada ano passado pela ONU - Organização das Nações Unidas. De acordo com a mesma, a quantidade de água existente no planeta é capaz de atender

a população mundial, no entanto não há mais condições para que ocorra o desperdício. Enquanto existem países cujo consumo de água *per capita* atinge aproximadamente 2333 m³ por ano [16], mais de um bilhão de pessoas não têm acesso à água potável para consumo, por esses motivos o problema deve ser tratado através de políticas de gerenciamento de recursos hídricos em todos os países do mundo.

O IFBA *campus* de Salvador desperdiça uma quantidade significativa de água e isso contribui para aumentar de forma preponderante os gastos da instituição. Esses recursos poderiam ser utilizados para resolver outras questões igualmente importantes como, por exemplo, compra de novos livros, computadores e outros equipamentos necessários ao bom funcionamento das atividades de ensino.

Este trabalho apresenta um sistema de monitoramento automático do uso da água no IFBA para efetuar o controle e gerenciamento do consumo. Este sistema permite supervisionar de forma automática quais são os locais em que estão utilizando água acima dos limites considerados normais e indica quais deles precisam de reparos estruturais.

O sistema consiste em uma plataforma de hardware (microcontrolador e sensor de consumo de água) e um software embarcado. O hardware adotado é de arquitetura aberta e de baixo custo e sua instalação pode ser feita por qualquer pessoa, mesmo aquelas com pouca experiência. O objetivo é permitir que o IFBA e outras instituições, empresas e residências possam utilizar o sistema sem despendar muitos recursos financeiros.

Além do software embarcado na plataforma de hardware, o sistema também contempla um software de interface web que permite acompanhar de forma visual os diversos pontos monitorados. Esse software contém diversos relatórios para apoiar as decisões de um gestor.

Para implementar este sistema, primeiramente foi feito um levantamento do estado da arte para os sistemas de monitoramento automatizado. Assim diversos trabalhos nas áreas de computação ubíqua, microcontroladores, sistemas de visualização de informação web foram estudados e ajudaram a definir as bases do sistema implementado. O próximo passo foi implementar o sistema usando um microcontrolador, um

^{*}Estudante do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS)

[†]Doutor em Ciência da Computação e Professor do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS)

sensor de vazão de água e toda a infra de sistemas web. Depois disso, foram realizadas a instalação e teste do sistema de monitoramento. Cada ponto coletou informações que foram enviadas para uma base de dados e posteriormente visualizadas através da interface web.

Este trabalho está organizado em cinco capítulos. Depois da introdução, o segundo capítulo trata do estado da arte relacionado às diversas tecnologias envolvidas na elaboração desta monografia, como objetivo de introduzir conceitos importantes da área da Computação em que o contexto do projeto está inserido. O terceiro capítulo apresenta os principais trabalhos relacionados ao sistema implementado. O quarto capítulo apresenta os detalhes da implementação executada. O quinto capítulo traz os testes realizados. A última seção apresenta as conclusões deste trabalho.

2. REFERENCIAL TEÓRICO

Nesta seção são apresentados os principais conceitos estudados, os quais forneceram subsídios ao desenvolvimento do projeto proposto.

2.1 Computação Ubíqua

O termo computação ubíqua [13] (do inglês, *ubiquitous computing*) foi um termo introduzido em 1988 por Mark Weiser, quem na época era diretor do Laboratório de Ciência da Computação da Xerox PARC, para referir-se a terceira era da computação moderna. A primeira era foi definida por *mainframes*, computadores de grande porte que era utilizado por muitas pessoas ao mesmo tempo. Após alguns anos, surgiu a era do PC, um computador pessoal de menor porte e custo utilizado por apenas uma pessoa. Apesar de ter surgido na década de 70, ainda é bastante usado na atualidade. A terceira era presente nos tempos atuais, computação ubíqua, é caracterizada pela crescente integração de dispositivos heterogêneos à vida cotidiana dos usuários sem que estes percebam que estão interagindo com diversos computadores. Em um artigo publicado na revista *Scientific American*, Mark Weiser explica a essência da computação ubíqua através de um conciso resumo: “*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*” Tradução: “*As tecnologias mais profundas são aquelas que desaparecem. Tecem-se no tecido da vida cotidiana, até que são indistinguíveis a partir dele.*” [21]

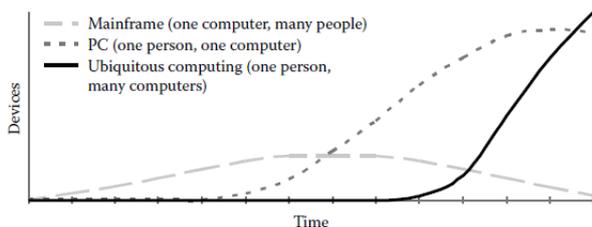


Figura 1: Eras da Computação [13]

Um diferencial das tecnologias ubíquas é o estilo de interação, que foge do modelo tradicional em que usuários devem estar sentados, usar um teclado e monitor, ou seja,

muita atenção é consumida impossibilitando os mesmos de perceberem o que acontece ao redor, além de não se movimentarem livremente enquanto usam o computador. Weiser acreditava que em um mundo *ubicomp* (como também é denominada a computação ubíqua), a computação estaria presente nos mais diversos dispositivos utilizados em atividades rotineiras como relógios, interruptores de luz, *smartphones* e em todos os lugares que quando integrados não é possível perceber que há vários computadores envolvidos. Tais dispositivos possuem diferentes sistemas operacionais, interfaces de rede e entradas e saídas, através dos quais o usuário final pode interagir, enquanto outros não são usados diretamente pelos usuários finais, sendo o caso dos sensores, por exemplo. Este contexto foi referenciado como *invisible computing*, computação invisível.

De acordo com [21], alguns autores utilizam os termos computação ubíqua e computação pervasiva como sinônimos. A computação pervasiva propõe que dispositivos esteja conectados o tempo todo, assegurando que o usuário possa acessar seus dados em qualquer momento e em qualquer lugar, sendo a única preocupação a disponibilidade do serviço. A computação móvel está diretamente ligada à mobilidade dos dispositivos, geralmente são pequenos e portáteis, que permite ao usuário utilizá-lo enquanto movimenta-se como por exemplo, *tablets* e *smartphones*. Apesar dos *notebooks* e *laptops* serem dispositivos de fácil transporte, requerem durante seu uso que o usuário esteja parado, portanto fazem parte da computação mais tradicional.

Como mostrado na figura 2, alguns autores consideram a computação ubíqua a interseção entre a computação móvel e pervasiva.



Figura 2: Computação Ubíqua: interseção entre computação pervasiva e computação móvel [13]

Pesquisas na área de Computação Ubíqua abordam sobre as tecnologias e infra-estruturas que viabilizam a implantação de aplicações ubíquas através de uma série de questões dentre elas: como projetar *hardwares* e sistemas operacionais para plataformas de sensores; como permitir que dispositivos encontrem uns aos outros e utilizem seus serviços, como permitir que sistemas envolvendo recursos limitados de processamento e energia funcionem; entre outros aspectos importantes. Conforme [6], geralmente as aplicações ubíquas recebem dados de sensores, de outros dispositivos provedores de serviços, gerenciam ações de usuários, oferecem suporte à mobilidade e utilizam informações de contexto para execução de tarefas.

Um sistema ubíquo propriamente dito possui um conjunto de requisitos, peculiaridades e desafios que influenciam na concepção, implementação, implantação e avaliação de seu

projeto, diferenciando dos demais tipos de sistemas desenvolvidos para PCs, como por exemplo:

1. *Resource-Constrained Devices* (Dispositivos de recursos limitados)
2. *Volatile Execution Environments* (Ambientes de execução voláteis)
3. *Heterogeneous Execution Environments* (Ambientes de execução heterogêneos)
4. *Fluctuating Usage Environments* (Ambientes de uso flutuantes)
5. *Invisible Computing* (Computação invisível)
6. *Security and Privacy* (Segurança e privacidade)

Através da figura 3, pode-se comparar o método científico com o método de desenvolvimento de sistemas *ubicom*. Assim como um experimento permite que uma hipótese seja refinada levando ciclicamente à novas experiências, o projeto de um sistema pode ser também melhorado. A implementação e avaliação de um sistema pode descobrir hipóteses de que podem ser testados experimentalmente e vice-versa. [13]

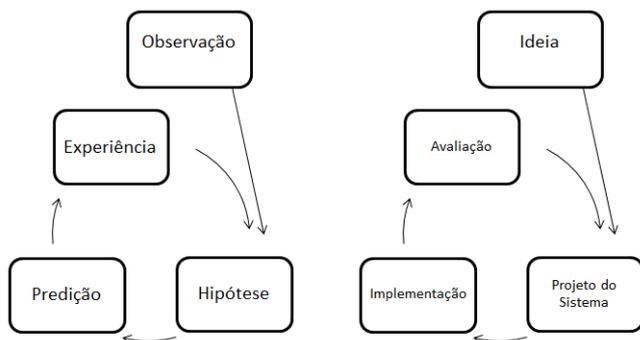


Figura 3: Metodologia para construção de sistemas ubíquos

[13]

2.2 Sistemas PIMS

PIMS *Plant Information Management System*

Na indústria, é importante a integração de informações de outros sistemas em apenas um único para fins de gerenciamento da produção. Dentro de tal contexto, sistemas PIMS são fortemente utilizados. Os sistemas PIMS são uma das ferramentas que oferecem suporte à sistemas de gerenciamento. De acordo com [19], *Plant Information Management Systems* são sistemas que adquirem dados de processo de diversas fontes como CLPs (Controladores Lógico-Programáveis), sistemas SCADA (*Supervisory Control and Data Acquisition*) e SDCD (Sistemas Digitais de Controle Distribuído), os armazenam num banco de dados histórico e os disponibilizam através de diversas formas de representação. PIMS nasceu na indústria de processos contínuos, mais especificamente a petroquímica e química, tornando-se

ferramenta fundamental à obtenção de uma visão unificada do processo. Atualmente novas ferramentas farão com que o PIMS seja indispensável também às indústrias de processos por bateladas e até manufatura.

Os sistemas PIMS consistem em integrar dados de processo de produção (valores de pressão, temperatura, modos de operação de equipamentos) e permitir transformar estes em informação e posteriormente em conhecimento. Dados do chão de fábrica provenientes de sensores e atuadores são transmitidos em rede e utilizados por sistemas supervisórios para representação do estado atual da planta. Esses dados são armazenados e utilizados posteriormente para análise de desempenho da unidade industrial, verificação de enquadramento em normas operacionais, gestão de recursos [20]. O engenheiro de processos químicos, por exemplo, pode visualizar dados de tempo real assim como dados históricos da planta, montar tabelas, gráficos de tendências que permitem tirar conclusões sobre o comportamento histórico e atual da planta por meio de comparações de dados de dias, meses ou até mesmo anos distintos. Um sistema PIMS é constituído por:

Historiador de processos Obtém dados de diversas fontes tais quais CLPs, sistemas SCADA ou SDCDs e armazenados em um banco de dados temporal. O conceito de banco de dados temporal será explicado na seção seguinte.

Interface gráfica Aplicação de visualização dos dados armazenados

Aplicações clientes Aplicações clientes complementares. Ex.: editor de texto, editor de cálculos

A arquitetura de um sistema PIMS possui três camadas, conforme imagem abaixo:

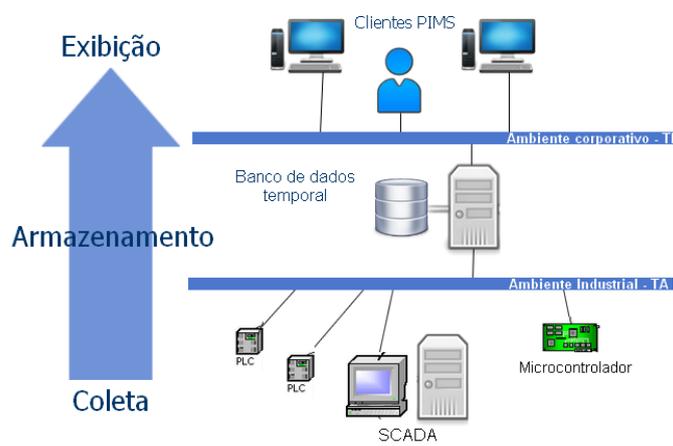


Figura 4: Arquitetura de sistemas PIMS

[19]

Na camada inferior de um PIMS, camada de coleta, os dados são obtidos a partir sistemas SCADA, CLPs e SDCDs aos quais os instrumentos estão conectados. A intermediária, camada de armazenamento, é responsável por manter os dados em um servidor de banco de dados. Finalmente, na camada

de exibição os os dados são consultados e exibidos em tempo real através de gráficos de tendência, por exemplo.

Apesar dos PIMS serem especializados no armazenamento de variáveis analógicas, outros tipos de dados também podem ser armazenados como por exemplo: variáveis discretas, texto e BLOBS *Binary Large Object* geralmente usados para armazenar fotos e sons. Devido ao grande volume de dados em sistemas industriais, tal armazenamento requer que esses dados tenham uma estrutura simples de forma a diminuir a complexidade de transferência, persistência dos dados e não onerar muito espaço em disco. Essencialmente as informações armazenadas no banco de dados histórico é uma lista de registros temporais com valores para os seguintes atributos:

Tag Cada fonte de dados citada anteriormente pode adquirir dados de mais de uma variável, sendo que cada variável está vinculada a um sensor. Tais variáveis recebem o nome de *tags* e representam uma importante unidade à camada de armazenamento. O tag consiste em um nome identificador, geralmente nome do instrumento correspondente na rede de automação.

Timestamp Momento em que o valor do tag foi assumido.

Valor

Qualidade do dado Tal atributo está relacionado à qualidade do dado mensurado pelo equipamento, a qual depende do estado de funcionamento do mesmo. Se o instrumento que obteve tal medida está calibrado ou não, se o dado estiver fora do intervalo de valores esperados são exemplos de condições que influenciam na qualidade do dado.



Figura 5: Importância dos dados para obtenção de informações

[19]

Conforme [19], a principal função do PIMS é concentrar quantidade de dados significativas de modo permitir transformar dados em informação em conhecimento. As informações geradas a partir dos dados históricos são bastantes significativas ao engenheiro de processo, pois assim este pode compreender as situações operacionais e compará-las com situações padrões, auxiliando na análise de eficiência da produção a partir dos resultados obtidos, levando o engenheiro a conhecer melhor a planta e criar melhorias no processo a fim torná-lo mais eficiente.

2.3 Banco de dados temporal

Para implantação de um sistema com base em armazenamento PIMS, pode-se utilizar a estratégia de modelagem Temporal Entidade Relacionamento. Essa modelagem baseia-se em variações do modelo relacional, adicionando associações e atributos temporais [20].

De acordo com [7], dados temporais em base de dados podem ser definidos em nível de atributos (*attribute timestamping*) ou entidades (*tuple timestamping*) presentes nas tabelas. No momento em que um atributo de uma tabela possui valor que variável no tempo, pode-se denominá-lo como um **atributo temporal**. Por exemplo, para o atributo valor, da tabela medida, existe uma associação temporal, pois geralmente o valor do consumo de água mensurado pelo sensor modificam com o passar do tempo. Quando um atributo é determinado temporal, existem alternativas para representar e reter o estados do mesmo ao longo de sua variação no tempo, seja pela utilização de instantes ou períodos, no caso deste projeto, a representação escolhida foi a baseada em instantes. Na imagem [2.2] a seguir é apresentada a tabela da base de dados que representa a entidade Medida populada com valores obtidos pelo sensor “BIBVAZ002”

tag	data_hora	valo	medidor_id
BIBVAZ002	10/07/2014 21:13:06	93,46	4
BIBVAZ002	12/07/2014 21:12:16	34,84	4
BIBVAZ002	20/07/2014 18:18:49	4,5	4
BIBVAZ002	20/07/2014 18:19:57	12,56	4
BIBVAZ002	20/07/2014 18:22:33	0	4

Figura 6: Dados dos valores medidos por “BIBVAZ002” em um modelo de representação baseada em instantes

O modelo do banco de dados definido para representar os dados temporais na base influencia no comportamento da mesma em diversos aspectos tais como consumo de espaço em disco, desempenho em consultas e desempenho em atualizações. Com a estrutura escolhida para este projeto, é possível alcançar menor tempo de resposta de uma consulta, devido à leitura ser realizada de maneira sequencial. Uma outra vantagem desde modelo é a facilidade de verificação dos atributos temporais, por exemplo, um medidor não pode obter valores distintos para a mesma data e hora. É possível garantir integridade dos dados em relação à essa restrição criando uma chave primária composta pelo identificador do medidor e *timestamp* do registro inserido.

2.4 Sistema de Medição

O sistema de monitoramento automático de consumo de água desenvolvido nesse projeto é constituído por um conjunto de dispositivos, também chamados de instrumentos para controle de processos, que desempenham diversas funções que asseguram o correto funcionamento do sistema. De acordo com a função desempenhada, tais dispositivos podem ser classificados e descritos conforme [4]:

- Sensor - Parte de uma malha ou de um instrumento que primeiro sente o valor da variável de processo.

- Transmissor - dispositivo que sente uma variável de processo por meio de um elemento primário e que produz uma saída cujo valor é geralmente proporcional ao valor da variável de processo. O elemento primário pode ser ou não parte integrante do transmissor.
- Controlador - Dispositivo que tem por finalidade manter em um valor predeterminado uma variável de processo.
- Conversor - Dispositivo que emite um sinal de saída padronizado modificado em relação à natureza do correspondente sinal de entrada.

No contexto industrial, um dispositivo de medição deve detectar algum parâmetro físico utilizado no processo tais como pressão, potência, temperatura e tensão. Além disso, é fundamental que tal dispositivo seja capaz de detectar precisamente variações no parâmetro detectado para fins de controle do processo, análise de perdas e de rendimentos, ressaltando que no contexto em que se insere esse projeto, o parâmetro vazão é mensurado a fim de estabelecer o monitoramento do consumo de água no Instituto Federal da Bahia. Atualmente, selecionar um medidor de vazão não é mais uma tarefa tão simples, pois há grande variedade de tecnologias disponíveis no mercado capazes trazer soluções e dúvidas ao mesmo tempo portanto, é primordial o claro entendimento da especificação da aplicação.

Os requisitos e particularidades do sistema influenciam significativamente para escolha do medidor de vazão. Para selecionar o mais apropriado deve-se levar em consideração propriedades específicas do líquido como pressão, temperatura, densidade, viscosidade, e características da tubulação onde será instalado o medidor, como o material, diâmetro, e direção da mesma. Após o claro entendimento dos requisitos, devem ser feitas avaliações da tecnologia, vantagens e desvantagens, limitações de funcionamento, custo, segurança e confiabilidade.

Assim como apresentado anteriormente, segundo Albuquerque (2005, p.96): os medidores de vazão são na realidade indiretos, pois medem outro tipo de grandeza que é transformada por um transdutor. Entre as variáveis medidas com frequência na indústria são: pressão, temperatura, densidade, viscosidade, umidade, possibilidade de mudança de fase e as impurezas. Com relação à geometria do duto, as variáveis de influência são: diâmetro ou secção reta, rugosidade, trecho reto e a direção (no caso de um tubo vertical). Não existe um só princípio de medição que possa atender a todas as condições. Alguns são limitados pelo diâmetro, outros pela temperatura, outros exigem longo trecho reto, etc. A escolha de um medidor para aplicação certa requer muita experiência em certos casos, pois além de ser adequado, de acordo com os primeiros critérios citados, há necessidade de verificar a precisão, a faixa de trabalho, a velocidade de resposta e finalmente o preço, que não é o menos importante, mas deve ser deixado como último critério.

2.5 Sensor de vazão

A vazão consiste em quantidade de fluido (líquido ou gás) que passa em um determinado ponto durante uma determinada unidade de tempo. A vazão também é denominada

vazão volumétrica, ao referir-se a um **volume** de um fluido escoado, ou vazão gravimétrica ao referir-se à **massa** do fluido escoado dada pela fórmula:

$$Q = \frac{V}{t} \quad \text{ou} \quad Q_M = \frac{M}{t}$$

Os sensores de vazão podem ser classificados em relação à grandeza medida, seja a mesma velocidade, força, tensão induzida e pressão diferencial.

Os principais medidores de vazão utilizados apresentam basicamente duas relações entrada/saída. Os de pressão diferencial, como por exemplo: placas de orifício, bocal de vazão e tubo de venturi possuem saída quadrática em relação à vazão, já outros medidores como o magnético, turbina, mássico e ultra-sônico possui saída linear em relação à vazão. Conforme avaliação dos principais critérios para seleção do medidor, o sensor selecionado foi o *G1/2 Water Flow Sensor* modelo POW110D3B fabricado pela *Seeed Studio*, do tipo turbina adequado para medir vazão da água, por conta das propriedades físico-químicas, por exemplo: fluido líquido de baixa densidade, não volátil à temperatura ambiente, não tóxico entre outras que estão de acordo com a especificação do sensor, além do baixo custo.

Tabela 1: Especificação do G1/2 Water Flow Sensor

Min. Tensão de funcionamento	DC 4.5 V
Max. Trabalho	15 mA de corrente (DC 5V)
Tensão de funcionamento	5V - 24V
Vazão Faixa	1 - 30L/min
Capacidade de carga	10 mA (DC 5V)
Temperatura de operação	80°C
Temperatura do Líquido	120°C
Umidade	35% - 90% RH
Pressão da água	2.0 MPa
Temperatura de armazenamento	-25°C- 80°C
Umidade de armazenamento	2% - 95% RH

O *G1/2 Water Flow Sensor* é um sensor do tipo turbina, cujo funcionamento pode ser explicado uma maneira geral da seguinte maneira, segundo [Jose Luiz Loureiro Alves]: “Inventado por Reinhard Woltman, o medidor do tipo turbina é um medidor de vazão exato e confiável, tanto para líquido quanto gases. Ele consiste num rotor com várias lâminas montadas num determinado ângulo e suspenso por um rolamento que o permite rolar livremente. A vazão do fluido é proporcional à velocidade de rotação da turbina. A velocidade de rotação pode ser detectada por dispositivos de estado sólido como sensores de relutância, capacitância, indutivos ou por sensores mecânicos.”

Em particular o *G1/2 Water Flow Sensor*, modelo da figura 7 é constituído por um corpo e válvula de plástico, um rotor e um sensor de efeito Hall. Ao alimentar o rotor, a corrente que passa pelo fio produz um campo eletromagnético detectado pelo sensor de efeito Hall. Como a turbina é rotacional, a água que passa faz com que a turbina gire e a corrente detectada gere pulsos cuja frequência é proporcional à velocidade do fluido.



Figura 7: G1/2 Water Flow Sensor modelo POW110D3B

[1]

Quando a água escoar pelo rotor, o mesmo rola e sua velocidade varia com a taxa de fluxo. A partir daí o sensor de *efeito Hall* gera um pulso correspondente ao sinal. De acordo com a especificação, a frequência do pulso (Hz) é de $7.5Q$, sendo Q a taxa de fluxo em L/min, que pode ser lida com precisão de 3%. A relação de vazão por frequência do sensor é dada por:

$$Vazao = \frac{frequencia \cdot pulso * 60}{7.5}$$

2.6 Microcontroladores e Arduino

Nesta seção serão apresentados conceitos introdutórios sobre microcontroladores e suas principais aplicações. Desta maneira, torna-se mais fácil a compreensão sobre uma importante tecnologia utilizada no desenvolvimento deste projeto, o Arduino, tecnologia sendo bastante utilizada por profissionais de diferentes áreas do conhecimento para desenvolvimento de projetos de tecnologia e inovação.

Conforme [10], um microcontrolador é definido “como um pequeno componente eletrônico, dotado de inteligência programável, utilizado para o controle de processos lógicos”. Os microcontroladores são descritos por [12] como circuitos integrados contendo em seu interior os componentes essenciais ao seu funcionamento, dependente apenas de fonte de alimentação externa.

É composto por CPU (*Central Processing Unit*), memória RAM (*Random Access Memory*) portas de E/S e, possivelmente em alguns casos, conversores analógico digital e módulos de temporização. Os microcontroladores são utilizados pelos mais diversos dispositivos eletrônicos que fazem parte do cotidiano, como por exemplo: automóveis, impressoras, calculadoras, micro-ondas. São também utilizados na indústria nos controladores de processos, inversores e balanças, além de outros meios. Consomem pouca energia e possuem capacidade de “hibernar” enquanto aguardam por acontecimento de um evento que o põe em funcionamento novamente, o que auxilia para economia de energia.

É muito comum confundir os termos microprocessadores e microcontroladores, que apesar de parecidos, há diferença entre eles. Microprocessadores são descritos por [12] como circuitos integrados que agrupam todos os componentes necessários

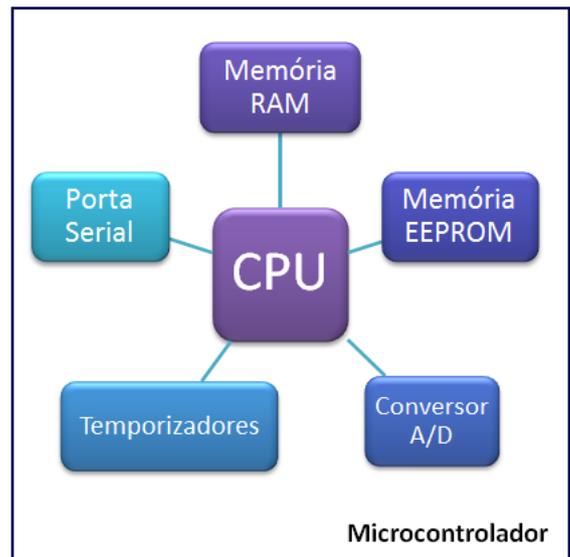


Figura 8: Componentes típicos de um microcontrolador

[9]

ao processamento de comandos de um programa, porém **não** possuem outros componentes como dispositivos de E/S e memórias. Em contrapartida, os microcontroladores possuem em sua arquitetura microprocessador (CPU), tem também outros importantes itens para que o sistema possa funcionar sem necessidade de componentes externos.

O Arduino foi criado em 2005 por Massimo Banzi e David Mellis na Itália com o objetivo de utilizar como ferramenta de aprendizagem de eletrônica e programação para estudantes de design, para que estes utilizassem em projetos de arte, interatividade e robótica.

Na obra *Getting Started with Arduino*, Massimo Banzi define o Arduino como uma plataforma de computação física *open-source* baseada numa simples placa com pinos de entrada/saída que implementa a linguagem *Processing*. É uma pequena, porém poderosa, placa constituída por um microcontrolador que pode ser facilmente programado através de uma interface USB e capaz de construir dispositivos eletrônicos e sistemas interessantes.

Trabalhar com a construção de um *hardware* demanda muito tempo e esforço, pois é necessário criar circuitos do zero, utilizar diversos componentes como resistores e capacitores, muitas soldagens. Utilizar a placa do Arduino abstrai todo esse processo de construção tornando-o mais simples, permitindo que pessoas das mais variadas áreas do conhecimento estejam aptas a construir seus projetos. Tal plataforma é hoje bastante utilizada no mundo inteiro por oferecer outras vantagens:

- É um ambiente multiplataforma, o qual pode executar nos principais sistemas operacionais, como Windows, Linux e Macintosh.
- É um *hardware open-source*, ou seja, o projeto do cir-

cuito está disponível de modo que caso alguém tenha interesse em criar a sua própria placa, basta comprar os componentes necessários.

- O custo do *hardware* é baixo, cerca de U\$ 35,00.
- É possível programá-lo através de um cabo USB ao invés de uma porta serial. Vale lembrar que os atuais computadores não possuem portas seriais.
- Possui ambiente de desenvolvimento com interface intuitiva para fácil uso



Figura 9: Arduino Uno [5]

2.6.1 Arduino UNO

O Arduino UNO é uma placa baseada no microcontrolador ATmega328. Possui 14 pinos de entrada/saída, (sendo que 6 podem ser usadas como saídas PWM), 6 entradas analógicas, um cristal oscilador, conexão USB, *plug* de conexão de energia, uma alimentação ICSP e um botão de reinicialização.

Tabela 2: Especificação do Arduino UNO [5]

Microcontrolador	DC ATmega328
Tensão operacional	5V
Tensão de alimentação (recomendada)	7-12V
Tensão de alimentação (limites)	6-20V
Pinos I/O digitais	14 (6 podem ser usadas como saídas PWM)
Pinos de entrada analógica	6
Corrente contínua por pino I/O	40 mA
Corrente contínua para o pino 3.3V	50 mA
Memória flash	32 KB (ATmega328)
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)
Velocidade de clock	16 MHz

O Arduino Uno pode ser alimentado através de conexão USB ou fonte externa de alimentação, seja através de uma fonte ou de uma bateria. A placa pode operar com uma alimentação de 6 a 20 volts, no entanto caso a alimentação seja inferior a 7 volts o pino 5V pode fornecer menos de 5 V e a placa apresentar instabilidade. Em casos em que a alimentação é superior a 12 volts, é possível que o regulador de tensão aqueça e se estrague. O recomendado é que a alimentação seja de 7 a 12 volts. [5]

A placa possui 4 pinos de alimentação, sendo estes:

VIN Através deste pino obtém-se a tensão de entrada aplicada no circuito mediante uso de alimentação externa.

5V Fonte de alimentação regulada gerada pelo regulador na placa. A placa pode ser alimentada por meio da tomada de energia (7-12V), conector USB (5V) ou pino VIN (7-12V). Não é recomendado fornecer tensão através dos pinos 5V e 3.3V, pois estes ignoram o regulador, podendo danificar a placa.

3.3V Alimentação de 3.3 volts proveniente do chip FTDI a placa. O chip FTDI da placa é responsável pela conversão de serial para USB.

GND - Pino terra

O ATmega328 tem 32KB de memória, sendo que 0,5 KB é utilizado pelo *bootloader*, programa carregado no microcontrolador. O *bootloader* fica ativo enquanto a placa é reinicializada e logo depois começa a rodar o programa que foi feito *upload* mais recente para placa. O comprimento e largura máximos do PCB (*Printed Circuit Board*) do Arduino Uno são respectivamente 6,86cm e 5,33 cm

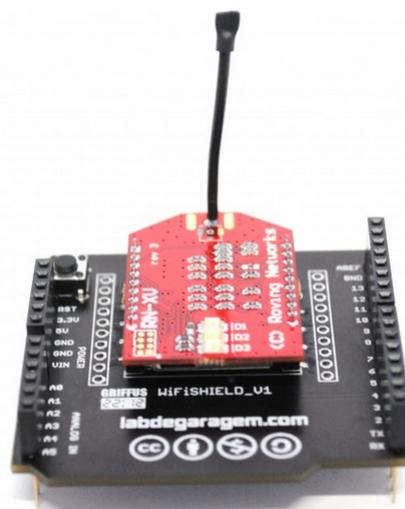


Figura 10: Wifly shield utilizado para conexão à rede sem fio [2]

Os conectores localizado na parte de cima da placa do Arduino podem ter duas finalidades: (1) encaixe de placas denominadas *shields* e (2) conectar fios do tipo *wire jumper*. As placas *shields* são capazes de executar funções específicas, como um display, driver para motores. Neste projeto desenvolvido foi utilizado o Wifly Shield, comercializada pela SparkFun [3], é um dispositivo independente que prepara o Arduino para estabelecer comunicação em rede sem fio padrão IEEE 802.11. Incorpora módulo Wifly RN-XV da Roving Networks, utiliza os pinos digitais de número 4 e 5 para comunicação serial e a alimentação do módulo é feita através do pino 3,3V. Através dessa conexão, os dados coletados pelo Arduino são enviados para persistência no servidor de banco de dados via *socket*.

3. SISTEMAS CORRELATOS

Nesta seção serão apresentados sistemas correlatos que foram utilizados como referência e conduziram a concepção deste projeto. Os principais projetos que serviram de apoio foram o Projeto Aguapura Vianet desenvolvido na UFBA (Universidade Federal da Bahia) e o projeto Rede de Estações Meteorológicas usando Arduino desenvolvido por Saulo Madalozzo na Universidade Regional Integrada do Alto do Uruguai e das Missões.

3.1 Programa Águapura Vianet (UFBA)

O Programa ÁGUAPURA [17] tem como objetivo reduzir as perdas e evitar desperdícios nas unidades da UFBA, além de difundir no meio universitário conceitos sobre uso racional da água de modo a conscientização da população quanto aos hábitos de consumir água de maneira sustentável.

Programa criado em 2001 consistia inicialmente em apenas uma equipe responsável pela manutenção corretiva e cadastramentos das unidades da UFBA (tanto de Salvador quando do interior da Bahia). Após reestruturação e revisão do programa, ações como inclusão de representantes nas unidades e criação do sistema ÁGUAPURA Via net acarretaram em progressos significativos. O sistema ÁGUAPURA Via Net permite inserção das leituras dos hidrômetros realizadas pelos representantes das unidades. Para executar tal procedimento, os representantes ou voluntários recebem login e senha para acesso ao sistema.

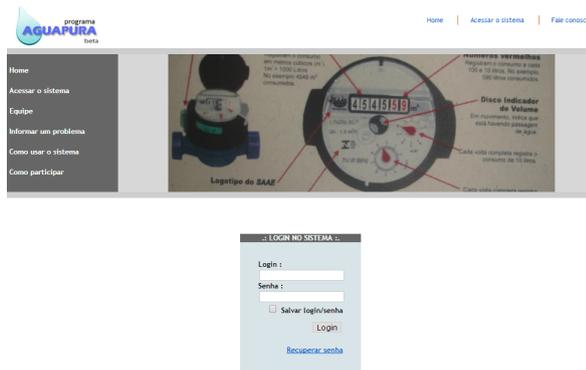


Figura 11: Tela de login do ÁGUAPURA Via Net

A partir de janeiro de 2005, com participação de voluntários, dados coletados dos hidrômetros são lançados no ÁGUAPURA Via Net e em maio de 2005, iniciaram-se ações de manutenção preventiva através de verificação nas instalações prediais.

A metodologia utilizada pelo Programa inclui cinco atividades, conforme [18]:

1. Levantamento do Sistema Hidráulico Predial
2. Monitoramento e Análise do Consumo de Água das Unidades
3. Detecção e Correção de Vazamentos Visíveis e Não Visíveis
4. Levantamento dos Hábitos dos Usuários

5. Utilização de Tecnologias de Processo e Produto para Racionalização do Consumo

Na atividade 1 levantou-se dados do sistema hidráulico de cada unidade, tipo e registro do medidor, e operacionalidade dos equipamentos. Na atividade 2, o monitoramento de análise do consumo de água das unidades consiste em acompanhar e monitorar as unidades cadastradas através do ÁGUAPURA Via Net. As leituras são realizadas *in loco* por voluntários das unidades para que este assim como a comunidade universitária perceba o seu consumo e conduza ao desenvolvimento de atitudes sustentáveis em relação ao uso da água. O acompanhamento é feito através de gráficos de consumo médio em cada unidade entre outras informações, sendo possível constatar vazamentos e anormalidades no consumo quando já é conhecido um padrão através do monitoramento diário. [18]

		Nov/2007	Dez/2007	Jan/2008	Fev/2008	Mar/2008	Abr/2008	Mai/2008
Arquitetura	D	4.24	4.27	5.28	10.29	6.67	4.65	4.26
	M	127.07	128.16*	158.30*	308.62*	200.12*	139.39*	127.88*
Belas Artes	D	4.42	3.47	1.48	2.17	2.27	2.72	2.52
	M	132.65	103.96*	44.27*	65.07*	68.00*	81.48*	87.75*
Biblioteca / Centro de Convivência	D	5.76	5.73	6.13	6.05	10.12	6.86	6.60
	M	172.70	171.98*	184.04*	181.40*	303.46*	205.73*	197.92*
Biologia	D	15.97	17.97	12.79	21.85	14.56	17.81	11.93
	M	479.13	539.17*	383.57*	655.36*	436.67*	534.25*	357.78*
Cantina da Escola Politécnica	D	1.94	1.39		1.56	1.86	1.84	1.74
	M	38.24	41.65*		46.79*	53.70*	53.33*	52.14*
Cantina de Veterinária	D	1.34	1.14		0.89	1.13	1.52	
	M	40.11	34.20*		26.60*	33.95*	45.51*	
Cantina do PAF I	D	4.47	5.57	3.75	3.49	4.98		
	M	134.18	167.08*	112.56*	104.70*	149.27*		
CEAO (Afro Oriental)	D							
	M							

Figura 12: Tela de monitoramento do consumo das unidades do Via Net

[18]

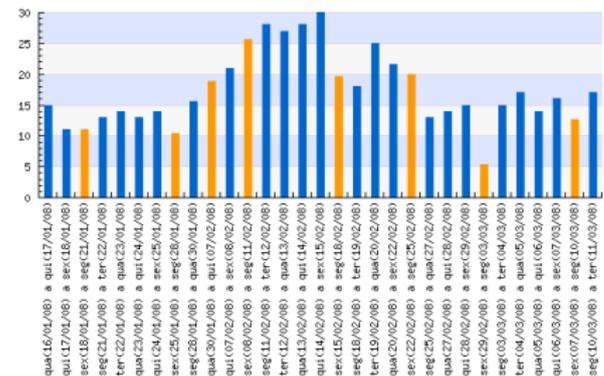


Figura 13: Gráfico de consumo médio diário em m^3 /dia

[18]

Na atividade 3 foram listados vazamentos visíveis e vazamentos invisíveis, sendo estes detectados através de geofone. Foi feita também manutenção dos prédios, com substituição de peças danificadas. Na atividade 4 foram aplicados questionários à população universitária, observação visual do comportamento do usuário para compreender melhor o perfil de consumo. Por fim, modificações nas instalações prediais (tecnologias de processo) foram realizadas para evitar desperdícios e controle dos gastos de água, como por exem-

plo: sistema de recirculação de água do sistema de resfriamento de destiladores, instalação de equipamentos redutores de consumo como torneiras de fechamento hidromecânico.

Algumas funcionalidades disponíveis no ÁGUAPURA Via Net são cadastro de setor, cadastro de ponto de consumo, cadastro de equipamento, mapeamento de ponto de consumo. Tais funcionalidades serviram de referência para o desenvolvimento do Portal SMA (Portal de Sustentabilidade e Meio Ambiente) que será apresentado na seção seguinte.

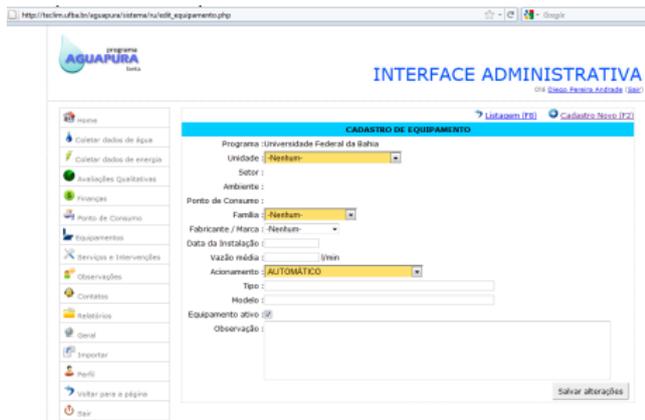


Figura 14: ÁGUAPURA Via Net - Tela de cadastro de equipamento

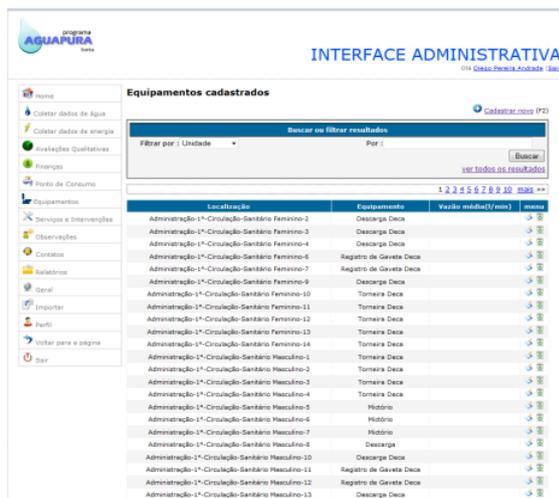


Figura 15: ÁGUAPURA Via Net - Tela de consulta de equipamentos cadastrados

3.2 Rede de Estações Meteorológicas usando Arduino

O projeto Rede de Estações Meteorológicas usando Arduino foi criado em 2013 com objetivo de criar uma base para uma rede de estações meteorológicas automáticas utilizando Arduino por meio do uso de *hardwares* simples e sensores largamente disponíveis no mercado, buscando uma boa relação entre custo, precisão e confiabilidade.

Este projeto é constituído por um conjunto de sensores os

quais coletam dados de variáveis que influenciam no tempo de maneira automática tais como: temperatura, umidade, pressão atmosférica, direção do vento e velocidade do vento. Os principais critérios de seleção das tecnologias foram custo

Arduino Ethernet (plataforma de *hardware* com microcontrolador e *Ethernet shield* para comunicação com em rede), sensor de temperatura, sensor de umidade, barômetros, pluviômetro são alguns dos recursos tecnológicos utilizados para o desenvolvimento do projeto. A imagem a seguir é apresentado um diagrama contendo as relações entre os componentes da estação meteorológica.

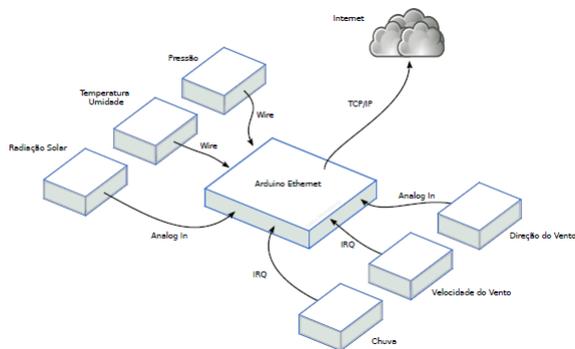


Figura 16: Comunicação entre os componentes da estação meteorológica

[14]

Além do *hardware* e *firmware* há também o software executado no servidor, o qual precisa trabalhar com grandes quantidades de dados. De acordo com [14], após 22 meses de funcionamento, o banco de dados da primeira estação tem 966.000 além de ocupar mais de 140MB de espaço em disco. Em menos de 30 dias o número de linhas passará para 1 milhão, ou seja, a base de dados desse projeto tende a crescer em pouco tempo, tornando o backup e manutenção custosos. A frequência do envio dos dados teve que ser bem modelada de modo a adquirir dados fidedignos à realidade em tempo real e não onerar processamento e espaço em disco, por isso, os dados são lidos a cada 30 segundos e após 5 minutos é calculada a média do valor que posteriormente é enviado ao servidor.

Neste tipo de sistema, pode-se ainda assim aumentar o intervalo de tempo de coleta e envio dos dados, pois as variações de temperatura, umidade entre outras variáveis climáticas não sofrem mudanças significativas em curto intervalo.

O *back-end* é composto por um script de recebimento dos dados e banco de dados. A persistência dos dados é executada por apenas um único arquivo implementado em PHP (acrônimo para "*Hypertext Preprocessor*") que recebe através de requisições GET advindas do Arduino Ethernet como parâmetro os dados que serão persistidos. Importante ressaltar que também são enviados informações para autenticação da estação, logo apenas são salvos dados de estações registradas. O tráfego dos dados pode ser facilmente compreendido através da imagem 17:

O módulo de visualização dos dados das estações meteo-

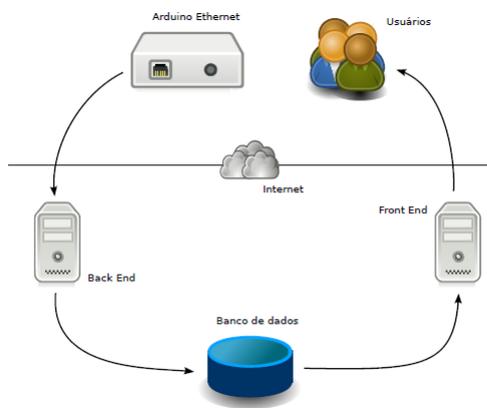


Figura 17: Tráfego de dados da estação meteorológica

[14]



Figura 18: Interface de visualização dos dados por meio de gráficos (Google Charts)

[14]

rológicas foi desenvolvida em PHP, utilizando o *framework* Twitter Bootstrap, capaz de gerar sites responsivos, ou seja, compatível com navegadores de *tablets*, *smartphones* e computadores com qualquer tamanho de tela. Nesta aplicação também foi utilizada a biblioteca Google Chart para gerar diversos tipos de gráficos, sejam estes interativos ou estáticos.

4. SISMA - SISTEMA INTEGRADO DE SUSTENTABILIDADE E MEIO AMBIENTE

Esta seção do artigo objetiva expor o funcionamento do sistema desenvolvido incluindo os seguintes tópicos:

- Requisitos: detalhamento dos requisitos funcionais e não funcionais
- Diagrama de detalhamento dos casos de uso e atividades

O objetivo do desenvolvimento do SISMA, Sistema Integrado de Sustentabilidade e Meio Ambiente, é a princípio

monitorar automaticamente o consumo de água no Instituto Federal de Educação, Ciência e Tecnologia da Bahia com a finalidade de realizar gerenciamento do consumo dos recursos hídricos e auxiliar na detecção de defeitos em componentes da instalação hidráulica do *campus*.

O IFBA é um instituição cuja maioria dos prédios construídos tem mais de cem anos, o que significa que por conta do desgaste do tempo, existe grande probabilidade de haver defeitos nos componentes hidráulicos e/ou ineficiência da utilização da água por alguns equipamentos já antigos (eg. condensadoras dos laboratórios de química), geralmente não perceptíveis por estarem localizados por baixo da terra ou interno às paredes. É, portanto, importante a utilização de um sistema como o proposto para minimizar o tempo entre ocorrência de danos e o reparo estrutural, que consequentemente reduz o desperdício.

O recurso financeiro o qual era destinado ao pagamento de contas de água com valores exorbitantes pode ser convertido em investimentos que visam a melhorias e crescimento da instituição através de reformas na infra-estrutura, execução de maior número de projetos, aquisição de livros para o acervo bibliográfico além de outros.

O SISMA é composto por uma plataforma de *hardware* (Arduino) e software embarcado de arquitetura aberta e baixo custo atualmente muito utilizada para elaboração de protótipos de produtos tecnológicos. O Arduino realiza leitura dos valores de vazão mensurados pelo sensor acoplado no mesmo e transfere para o servidor de banco de dados por meio de conexão à rede *wireless* para finalmente a aplicação *web* denominada "Portal SMA" disponibilizar as informações dos pontos monitorados aos usuários.

O Portal SMA permite acompanhar os pontos monitorados de forma visual, por meio de gráficos e tabelas contendo informações de consumo de água, que podem ser visualizados por período ou pela localização do medidor escolhida pelo usuário. Quando necessária uma análise mais minuciosa de como os recursos hídricos são utilizados dentro da instituição, o sistema permite também gerar diversos relatórios para apoiar às decisões relacionadas à gestão de recursos financeiros. Na atual versão do Portal SMA não é possível simular fatura, no entanto esta funcionalidade já foi planejada para próxima versão. Através da simulação de fatura o usuário poderá prever o valor da fatura enviada mensalmente pela concessionária de serviços de saneamento básico (Embasa), informando qual o montante deverá ser pago em decorrência do consumo.

Há uma seção no Portal SMA com conteúdo informativo sobre questões relacionadas à sustentabilidade e preservação do meio ambiente, além de dicas de boas práticas que visam a melhorar os eco-indicadores e conscientização da população. De maneira geral, podemos observar algumas das funcionalidades disponíveis no sistema através da tabela 3 com requisitos funcionais e não funcionais do sistema 4, além do Diagrama de Casos de Uso no próximo tópico.

4.1 Casos de Uso

Nesta seção, serão descritos detalhadamente os casos de uso exibidos no diagrama a seguir. Serão apresentados o passo

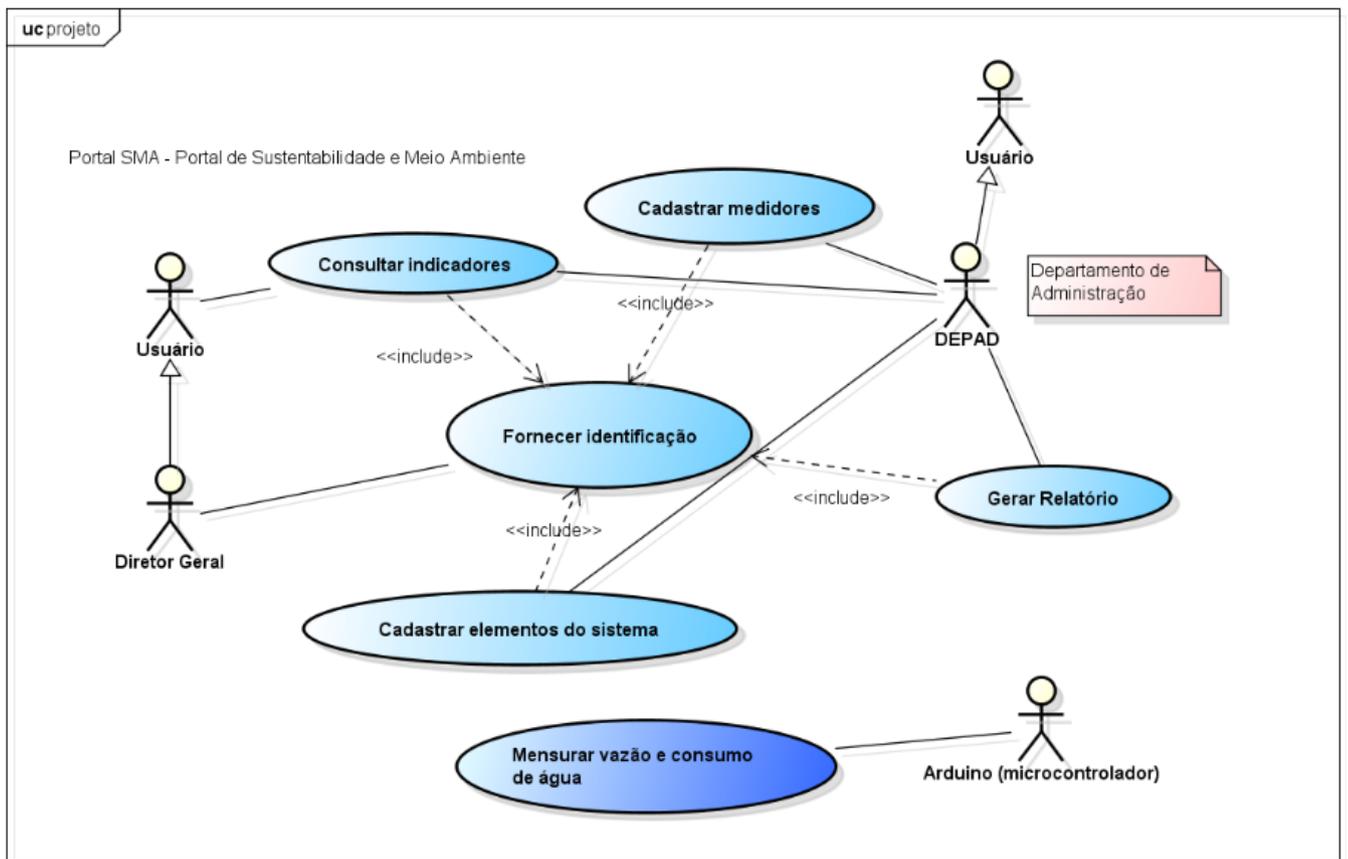


Figura 19: Diagrama de casos de uso do Portal SMA

a passo necessário para o usuário operar o sistema e obter os resultados esperados.

O sistema será utilizado por dois perfis de usuários:

1. Usuários administradores: Correspondem aos integrantes do DEPAD (Departamento Administrativo), CISA (Comissão Interna de Sustentabilidade do IFBA) e o Diretor Geral da instituição. De maneira geral, são as pessoas que são responsáveis pela manutenção do sistema. Os casos de uso [UC01], [UC02], [UC03] e [UC04] descritos em anexo apenas é realizado por este tipo de usuário.
2. Usuários de maneira geral, o que envolvem quaisquer pessoas de qualquer lugar do mundo. Os indicadores registrados estarão disponíveis para todos poderem visualizar sem a necessidade de cadastrar-se, ou seja, toda informação é transparente.

4.2 Arquitetura

A metodologia de desenvolvimento deste projeto teve como referência a metodologia de desenvolvimento de sistemas ubíquos apresentada no capítulo 2. Conforme diagrama da figura 3, o projeto consistiu em quatro principais fases: ideia, projeto do sistema, implementação e avaliação. A partir da observação de um problema, alto consumo de água no Instituto Federal da Bahia, foi proposta a ideia do sistema de

monitoramento como solução. A partir da pesquisa de sistemas correlatos, tomaram-se algumas características como modelo e nesta fase de idealização, foram estabelecidos os principais requisitos e como estes devem funcionar de modo a atingir o principal objetivo: monitoramento automático de água. Foram criados protótipos de baixa fidelidade através de simples rascunhos em papel, onde foram expostos e discutidos o cenário no qual o sistema se insere e como os usuários interagem com o mesmo. Durante a fase de projeto, foi realizada pesquisa das tecnologias e plataformas utilizadas para desenvolvimento do projeto, definição da arquitetura e padrões de projetos.

A implementação do projeto foi dividida em três partes de acordo com os módulos constituintes do sistema: aplicação embarcada do Arduino, serviço de persistência dos dados e a aplicação web para visualização dos dados. A aplicação embarcada é responsável pelo tratamento dos dados adquiridos pelo sensor acoplado ao Arduino, são efetuados o cálculo de volume gasta a partir da vazão mensurada e transmissão dos valores pela rede *wireless*. A comunicação entre os módulos é realizada via *socket*, interface que possibilita a comunicação entre sistemas implementados em diferentes linguagens de programação e que estão em máquinas diferentes. No serviço de persistência são tratados os dados recebidos e é executada inserção dos valores em uma base de dados para posterior visualização das informações no Portal SMA, aplicação web desenvolvida sobre a plataforma Java

EE(Enterprise Edition).

Os artefatos produzidos e tecnologias utilizadas na fase de implementação serão detalhados nas próximas subseções.

4.3 Aplicação embarcada

Antes de iniciar codificação da aplicação embarcada, foi necessária fazer a montagem do circuito para que o sistema pudesse funcionar devidamente. Componentes utilizados:

- Wifly Shield [3]
- Arduino Uno
- Sensor de vazão modelo POW110D3B
- Resistor de 10K Ohm
- *Protoboard* e fios

A instalação do *hardware* foi um procedimento relativamente simples de ser realizado. O sensor de vazão possui 3 fios de cores diferentes: preto, amarelo e vermelho. O pino vermelho é referente à alimentação logo, foi ligado ao pino 5V, o fio preto é ligado ao fio terra e por fim, o fio amarelo é responsável pelo envio do sinal, que é captado pelo pino 2. A montagem foi realizada de acordo com esquema da figura a seguir:

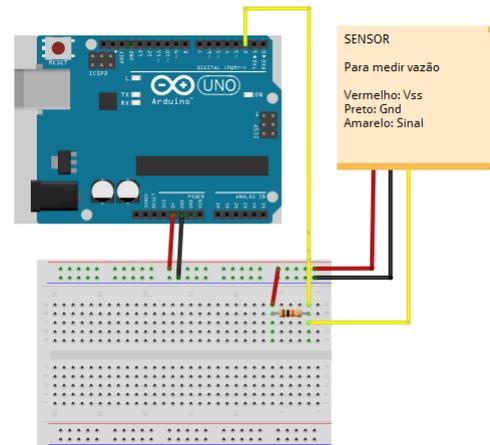


Figura 20: Esquema de instalação dos componentes para coleta de dados do sensor

A aplicação embarcada para coleta e envio dos dados obtidos pelo sensor foi criada com uso da IDE do Arduino, onde é feito todo o código também denominado *sketch*, em linguagem cuja sintaxe é semelhante a da linguagem C, que diz à placa o que deve ser executado. Após compilação, quando ocorrida com sucesso, já é possível realizar *upload* no Arduino, e quando este procedimento é feito, a IDE envia o *sketch* para a memória da placa que posteriormente irá executá-lo. Uma vez que o código foi salvo na memória do Arduino, esse estará disponível até outro *sketch* ser enviado, mesmo se a placa for desligada ou reiniciada [15].

Para verificar o valor obtido pelo sensor de vazão, a lógica abaixo foi utilizada e consiste determinar valor da vazão a

Tabela 3: Requisitos Funcionais do SISMA

Código	Nome	Descrição
[RF001]	Calcular Volume	O sistema deve calcular volume gasto a partir do valor de vazão de água coletado.
[RF002]	Gerar Gráficos	O sistema deve gerar gráficos (de linha ou de barras) para visualização dos dados coletados.
[RF003]	Consultar indicadores	O sistema deve permitir consulta de dados de acordo com filtro desejado pelo usuário.
[RF004]	Gerar relatório	O sistema deve permitir ao usuário gerar relatório dos dados consultados de acordo com filtro selecionado. Dados são exportados em arquivo de extensão ".xls"
[RF005]	Persistir dados	O sistema deve persistir dados coletados de medidores cadastrados no sistema.
[RF006]	Cadastrar Medidor	O sistema deve permitir ao usuário administrador o cadastro de medidores.

Tabela 4: Requisitos Não Funcionais do SISMA

Código	Descrição	Tipo
[RNF001]	Gráfico de linha deve ser gerado para mais de 15 registros	Interface
[RNF002]	Os valores de consumo de água devem ser enviados e persistidos em até uma hora	Performance
[RNF003]	Devem apenas ser enviados valores de consumo que não se repetem.	Performance

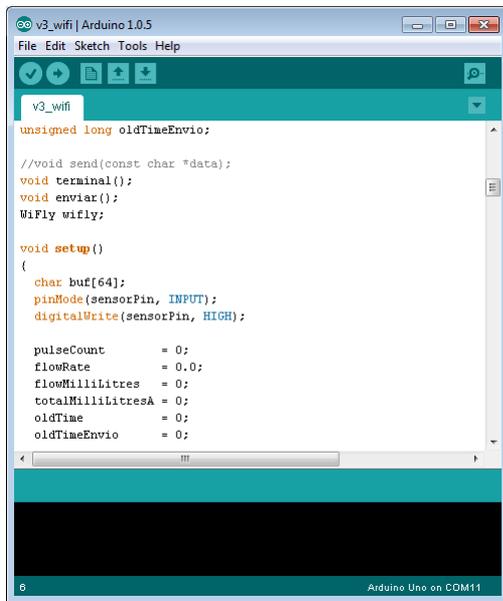


Figura 21: Aplicação embarcada desenvolvida para coleta e envio dos dados

partir da quantidade de pulsos gerados pelo sensor de efeito Hall e realizar cálculo da vazão da água com 3% de precisão.

```

// the loop() method runs over and over again,
// as long as the Arduino has power
void loop ()
{
  NbTopsFan = 0; //Set NbTops to 0 ready for calculations
  sei(); //Enables interrupts
  delay (1000); //Wait 1 second
  cli(); //Disable interrupts
  Calc = (NbTopsFan * 60 / 5.5); //(Pulse frequency x 60) / 5.5Q, = flow rate
  in L/hour
  Serial.print (Calc, DEC); //Prints the number calculated above
  Serial.print (" L/hour\r\n"); //Prints "L/hour" and returns a new line
}

```

Figura 22: Lógica para coletar valor de vazão mensurado pelo sensor [1]

4.4 Banco de dados

O modelo do banco de dados da aplicação foi construído com base nos conceitos de modelagem temporal de banco de dados explicada na seção 2.3.

4.5 Módulo de visualização - Portal SMA

O Portal SMA (Portal de Sustentabilidade e Meio Ambiente) foi desenvolvido sobre a plataforma Java EE (*Enterprise Edition*), visto que a implementação *web* torna mais fácil para o usuário utilização da aplicação já que esta não necessita estar instalada em uma estação de trabalho assim como o usuário não precisa estar limitado fisicamente a uma máquina, é possível acesso por meios de qualquer máquina e dispositivos móveis como *tablets* e *smartphones*. A plataforma Java EE permite integração com os *frameworks* JPA (*Java Persistence API*) e JSF (*Java Server Faces*) ao ambiente de desenvolvimento e por ser voltada para aplicações *web*, é essencial um servidor de aplicação, que no caso desse projeto é usado o Apache Tomcat. Além das tecnologias citadas anteriormente, fazem parte do ambiente de desenvolvimento também o PostgreSQL como SGBD (Sistema

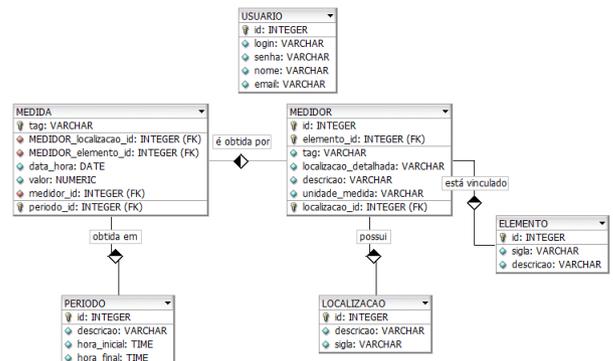


Figura 23: Modelo físico do banco de dados

Gerenciador de Banco de Dados), *Hibernate*, EJB (*Enterprise Java Beans*) e *Primefaces*.

A arquitetura do Portal SMA foi implementada em três camadas apresentando padrão arquitetural MVC (Model-View-Controller/Modelo-Visão-Controle) [25], o qual promove independência dos dados manipulados e as interações do usuário que dependem destes dados. Na camada de *View* são encapsuladas a saída da representação dos dados, no contexto do projeto, as páginas JSF que exibem tabelas e gráficos gerados através dos *frameworks* JSF e *Primefaces* integrados. A camada *Model* contém classes base relacionadas ao negócio da aplicação, como por exemplo as classes *Medidor* e *Medida*.

De acordo com [8], a camada *View* representa a parte da aplicação que interagem com usuário, as telas, que podem ser formadas por arquivos HTML, CSS e JavaScript, por exemplo, ou seja, estão nessa camada apenas os componentes para apresentação dos dados.

A *Controller* é responsável por receber as requisições do usuário quando interagem com a *view*, montar os objetos correspondentes e passá-los à camada *Model* para executar o que foi solicitado a fazer (persistência em banco de dados, por exemplo).

A camada *Model* é responsável pela lógica e inteligência do sistema: validação de negócio, cálculos, integrações e processamento. Com tais lógicas desacopladas da infra-estrutura da aplicação, garante maior flexibilidade já que é possível reaproveitá-las. Exemplo: pode-se reaproveitar módulo de cálculo de taxas do sistema de uma concessionária de uma aplicação *web* em um módulo *Desktop* de sistema com mesmo tipo de funcionamento. As interações entre as camadas ocorrem como apresenta a figura:

4.6 Principais Funcionalidades

O Portal SMA, deve ser acessado por meio de um navegador *web* para visualização dos indicadores de consumo de água, sendo assim, as principais funcionalidades que devem estar devidamente disponíveis são: gerar relatório (exportar dados) e gerar gráfico.

Para ter acesso às informações, sejam em forma de gráficos ou relatório, o usuário deve selecionar os filtros a fim de

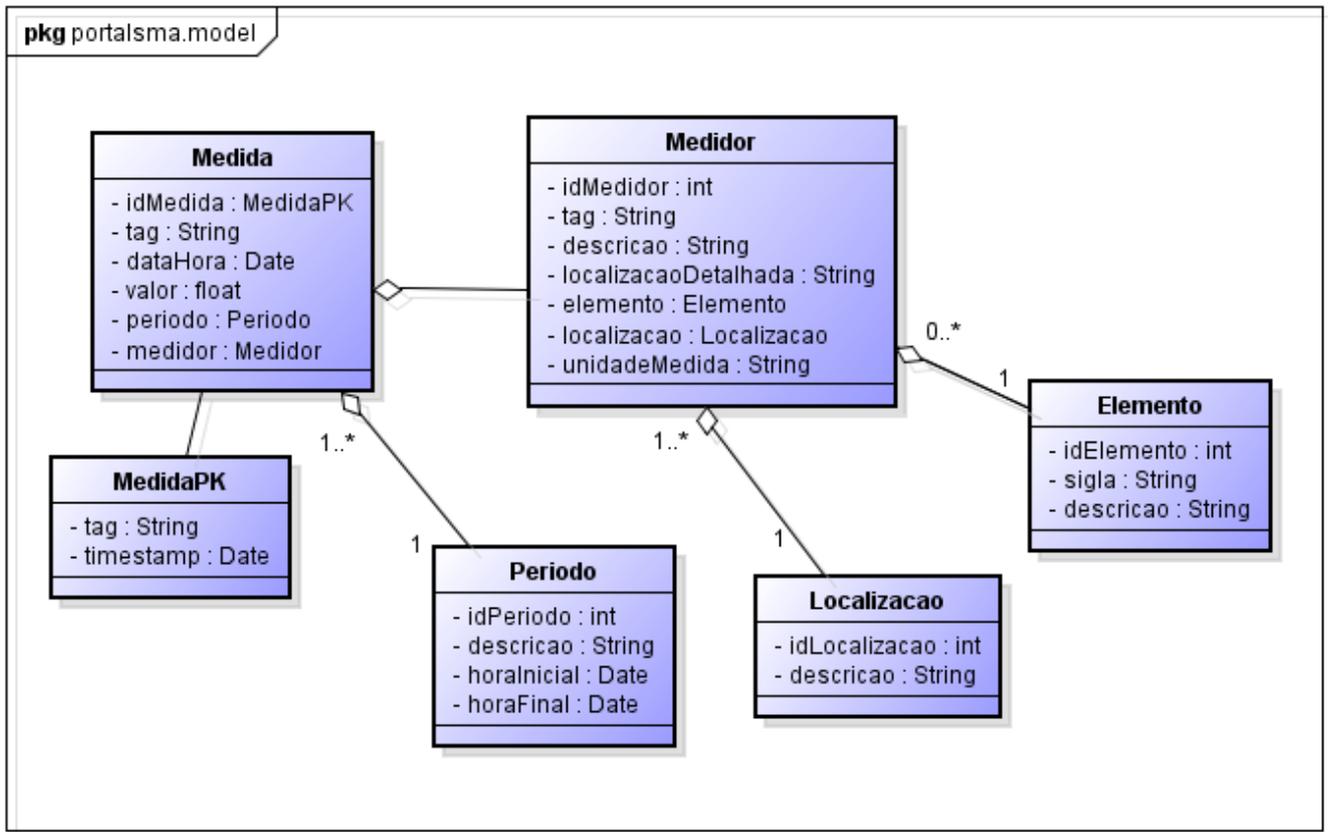


Figura 24: Diagrama de Classe da camada de negócio

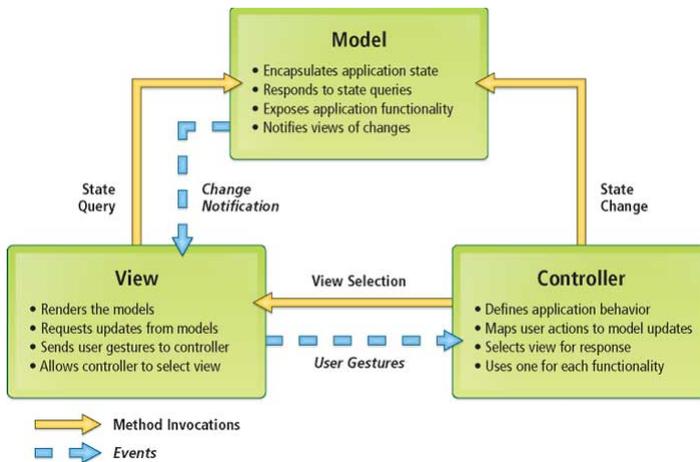


Figura 25: Padrão MVC (*Model-View-Controller*) [11]

obter a informação desejada. Os principais filtros são:

1. Medidor - Nome do medidor do qual deseja obter medida. Não é filtro de preenchimento obrigatório.
2. Data inicial e data final - Período o qual deseja visualizar o consumo de água.

3. Localização - Qual ponto de coleta deseja verificar. Exemplo: Bloco K, Biblioteca.
4. Período - Intervalo de tempo dentro de um dia com hora inicial e final definidos. Exemplo: madrugada (00:00 até 05:59), manhã (06:00 até 11:59), tarde (12:00 até 17:59) e noite (18:00 até 00:59).
5. Elemento - Elemento da instalação hidráulica ao qual o medidor está vinculado. Exemplo: chuveiro, torneira e condensadora.

Nem todos os campos descritos acima são de preenchimento obrigatórios, apenas são data inicial, data final e localização. Caso um medidor tenha sido selecionado, as opções de localização e elemento estão desabilitadas já que tal medidor já tem esses valores bem definidos. Ao clicar no botão "Gerar Gráfico", o gráfico é exibido na tela, caso haja dados para o(s) filtro(s) selecionado(s) pelo usuário 26.

O mesmo fluxo explicado anteriormente vale para a funcionalidade de gerar relatório, o que muda, é apenas a maneira como os dados são exibidos sendo que neste caso, estão é forma de tabela.

Importante lembrar que para acessar às funcionalidades administrativas do portal como cadastro de medidor, período e elementos da instalação hidráulica, o usuário deve estar autenticado.

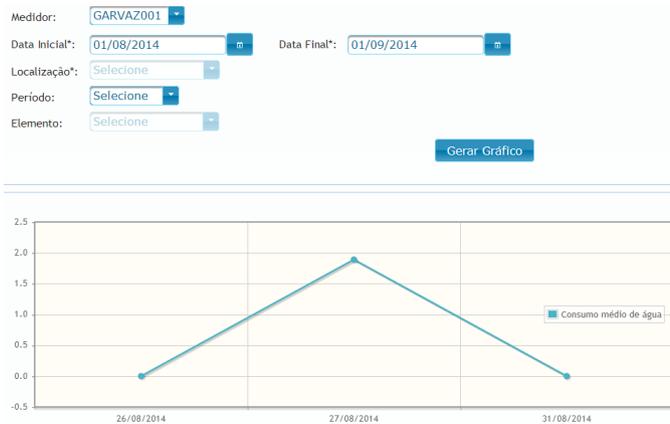


Figura 26: Tela de gerar Gráfico do Portal SMA



Figura 27: Tela de gerar Relatório do Portal SMA

5. TESTES E RESULTADOS

Nesta seção são descritas os principais a método utilizado para realização dos testes do sistema e principais resultados obtidos. Como já explicado na seção 4.2, a implementação da aplicação foi dividida em 3 partes, tais quais: (1) aplicação embarcada, (2) interface de comunicação e finalmente (3) aplicação *web*. De acordo com funcionamento do sistema, a plataforma do Arduino conecta-se a um servidor e envia valores mensurados via *socket* em intervalo de tempo pré-configurado. Após completada esta atividade, é verificado se o medidor é cadastrado para prosseguir com a persistência, em caso afirmativo. Para testar o SISMA, inicialmente foram realizadas verificações para cada parte desenvolvida com dados fictícios ou em condições que não condizem fidedignamente com o cenário real, que após validadas foram integradas a um único tipo de teste. Nesse caso, após implementação foram observados se valores eram medidos corretamente, enviados de acordo com tempo configurado e se os dados dos gráficos gerados refletem com os dados coletados e persistidos.

Após validação de cada etapa de desenvolvimento, foi criado um procedimento de teste integrando todas os componentes constituintes do sistema. No ambiente montado para execução deste procedimento (imagem 28), foi necessário uti-

lizar um funil ligado à uma mangueira conectada ao sensor de vazão, vaso graduado para ter estimativa do valor de consumo que deveria ser indicado pelo Arduino e serviços para recebimento persistência estarem disponíveis.



Figura 28: Ambiente de teste

Foram realizados testes utilizando 1, 2 e 4L de água e diferentes fatores de calibração do sensor de vazão. De acordo com sua especificação [1] o fator de calibração utilizado no cálculo de vazão é 7.5, no entanto ao verificar código-fonte apresentado nesta mesma fonte, o fator usado para realizar os cálculos foi de 5.5, portanto foram testados através testes dois parâmetros de modo a avaliar qual é o mais preciso. Lembrando que conforme especificação do sensor, já existe uma margem de erro associada de $\pm 3\%$.

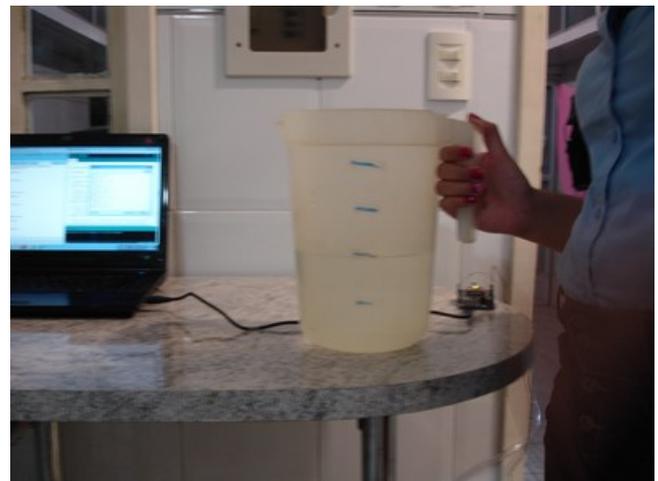


Figura 29: Ambiente de teste - Pote graduado usado para testes

O teste consistiu em preencher vaso graduado, conforme figura 29, contendo a quantidade desejada para teste (1, 2 ou 4 litros), inserir todo volume no funil para que a água desça por gravidade passando pelo sensor de vazão e descartando no balde logo abaixo do sensor. Importante ressaltar que para efetuar este procedimento, as aplicações já devem estar em execução. Além disso, o vaso da imagem acima é

devidamente graduado, no entanto a graduação é transparente, não visível na imagem, por conta disso foram feitas marcas manualmente com pincel para cada litro. Verificar figura 30.



Figura 30: Ambiente de teste - Representação simplificada de um sistema hidráulico

Após execução dos testes foram obtidos os seguintes resultados:

Tabela 5: Resultados obtidos com fator de calibração do sensor igual a 5.5

Quantidade de água despejada	Consumo de água indicado
1 Litro	1.1L
2 Litros	2.4L
4 Litros	4.8L

Tabela 6: Resultados obtidos com fator de calibração do sensor igual a 7.5

Quantidade de água despejada	Consumo de água indicado
1 Litro	0.8L
2 Litros	1.7L
4 Litros	4.14L

Nenhum dos dois valores do fator de calibração foram satisfatórios para obtenção de medidas mais precisas, tendo como referencial a quantidade de água despejada. Realizou-se, então, execução do mesmo teste considerando o fator igual a 6.5, média entre os dois fatores apresentados na especificação do sensor, e os resultados obtidos foram:

Apesar de alguns desvios, que de certa forma apresenta linearidade, foi verificado que o fator de calibração utilizado no cálculo capaz de trazer valores mais próximos do esperado

Tabela 7: Resultados obtidos com fator de calibração do sensor igual a 6.5

Quantidade de água despejada	Consumo de água indicado
1 Litro	1.12L
2 Litros	1.97L
4 Litros	3.60L

foi este último, concluindo que é o mais apropriado para ser usado.

6. CONCLUSÕES E TRABALHOS FUTUROS

No atual cenário, a escassez da água é um problema grave que impacta o meio ambiente e a vida da população, já que tal recurso é utilizado para as pessoas realizarem atividades rotineiras (cozinhar, tomar banho e claro ingerir água), indústrias de bens de consumo e produção de energia elétrica é de extrema importância economia desse recurso.

Para promoção de um desenvolvimento sustentável, projetos como este, exercem um papel fundamental para preservação do meio ambiente. Através do Sistema Integrado de Sustentabilidade e Meio Ambiente, a Comissão Interna de Sustentabilidade do IFBA já possui um ponto de partida para condução de melhorias e progressos dentro da instituição adotando através de políticas de conscientização e entre outras ações advindas de análises durante uso do sistema.

Apesar dos principais recursos da aplicação terem sido desenvolvidos, existem ainda pontos de melhorias. É interessante que novas funcionalidades de configuração sejam criadas, para que o administrador tenha maior flexibilidade e configurar o sistema de acordo com as suas necessidades. Outras funcionalidades como também envio de e-mails de alerta em casos do consumo de água exceder o limite configurado, para que assim seja mais rápido e eficiente o diagnóstico de vazamentos, por exemplo, criação de tela inicial de fácil visualização de informações gerais (*Dashboard*).

7. REFERÊNCIAS

- [1] *G1/2 Water Flow sensor*.
- [2] *Laboratório de Garagem - Wifi Shield*.
”<http://arduino.cc/en/Main/ArduinoBoardUno>”.
- [3] *Wifi Shield*.
- [4] J. L. L. Alves. *Instrumentação, Controle e Automação de Processos*. LTC Editora, Rio de Janeiro, RJ, 2005.
- [5] Arduino. *Arduino Uno*.
”<http://arduino.cc/en/Main/ArduinoBoardUno>”.
- [6] C. Batista, G. Alves, E. Cavalcante, L. Frederico, and T. Batista. Monitoramento de metadados para computação ubíqua. In *XXXIX Seminário Integrado de Software e Hardware (SEMISH 2012)*, Curitiba, PR, Brasil. *Anais do XXXII Congresso da Sociedade Brasileira de Computação (CSBC 2012)*. Porto Alegre, S, Brasil: SBC, 2012.
- [7] T. Cassol. Um estudo sobre alternativas de representação de dados temporais em banco de dados relacionais. Master’s thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2012.
- [8] G. Cordeiro. *Aplicações Java para web com JSF e*

JPA. Casa do Código, São Paulo, SP, Brasil, 2012.

- [9] B. B. de Oliveira Net; Priscila de França Monteiro; Sandro Lino Moreira de Queiroga. Aplicabilidade dos microcontroladores em inovações tecnológicas. *VII CONNEPI (Congresso Norte Nordeste de Pesquisa e Inovação)*. <http://propi.ifto.edu.br/ocs/index.php/connepi/vii/paper/viewFile/2433/2526>.
- [10] D. J. de Souza. *Desbravando o PIC: Ampliado e atualizado para PIC16F628A*. Érica, 2005.
- [11] M. DeMeritt. A compelling case for asp.net mvc. 2009. "http://www.esri.com/news/arcuser/0609/files/aspnetmvc.pdf".
- [12] R. KERSCHBAUMER. Microcontroladores. 2013.
- [13] J. Krumm. *Ubiquitous Computing Fundamentals*. CRC Press, Redmond, Washington, U.S.A, 2010.
- [14] S. Madalozzo. Rede de estações meteorológicas automáticas usando arduino. Graduação em ciência da computação, Universidade Regional Integrada do Alto Uruguai e das Missões - Campus Erechim, Erechim, RS, 2013.
- [15] c.-f. o. A. Massimo Banzi. *Getting Started With Arduino*. O'Reilly, CA, USA., 2nd edition, 2011.
- [16] M. Mekonnen and Hoekstra. National water footprint accounts: the green, blue and grey water footprint of production and consumption. 2011.
- [17] A. K. NAKAGAWA. *Caracterização do consumo de água em prédios universitários: o caso da UFBA. 2009. 183 f.* PhD thesis, Dissertação (Mestrado Profissional em Gerenciamento e Tecnologias Ambientais no Processo Produtivo)–Departamento de Engenharia Ambiental–DEA, Universidade Federal da Bahia, Salvador-BA, 2009.
- [18] A. K. Nakagawa, A. Kiperstok, and K. P. O. Esquerre. Águapura-programa de uso racional da água da ufba: Metodologia e resultados.
- [19] C. SEIXAS FILHO. Pims -process information management system Ū uma introdução.
- [20] G. P. G. Thiago C. Jesus, Breno C. Ramo. Desenvolvimento de um sistema de aquisição, armazenamento e publicação de dados baseado em sistemas pims. *XIX CBA (Congresso Nacional de Automática)*, 2012.
- [21] W. Vieira. Um framework para aplicações sensíveis ao contexto em dispositivos móveis: Framework camobile. Graduação em sistemas de informação, Universidade do Estado da Bahia, Salvador, BA, 2012.
- [22] F. Villela. Desperdício de água tratada pode chegar a 50% nas grandes cidades só com vazamentos da rede. 2011.

8. ANEXO

Descrição dos casos de uso do Portal SMA.

8.1 [UC01] Fornecer identificação

Esta funcionalidade é apenas utilizadas pelos usuários administradores do sistema com a finalidade de permitir acesso às outras funções do sistemas tais como cadastrar medidores, simular fatura, gerar relatório e cadastrar elementos do sistema.

Pré condições: O usuário deve estar devidamente cadastrado

no sistema.

Fluxo de eventos

1. Na tela inicial do Portal SMA, há os campos "Login" e "Senha" de preenchimento obrigatório.
2. Preencher o campo "Login" e "Senha";
3. Clicar no botão "Fazer Login";
4. Caso os campos estejam preenchidos corretamente, o *dashboard* do portal é exibido;
5. O caso de uso é encerrado

8.2 [UC02] Cadastrar medidores

Esta funcionalidade permite o usuário cadastrar medidores implantados para visualizar no portal os dados mensurados pelo mesmo.

Pré condições: O usuário deve ser administrador e estar logado no sistema.

Fluxo de eventos

1. Na tela inicial do Portal SMA, no menu superior há a opção "Cadastrar Medidores". O usuário deve selecionar tal opção.
2. Usuário deve obrigatoriamente preencher todos os campos para efetuar cadastro:

TAG do medidor Corresponde a texto de até 12 caracteres que serve como identificador do medidor, para facilitar a visualização das medições de maneira geral. O preenchimento deste campo deve obedecer o padrão: Cidade(3) + bloco(1) + andar(1) + sala + elemento(3) + número(2). Exemplo: SSAK302CH01 = SSA (Salvador) + K + 3 (terceiro andar) + 02 (sala 2) + CH (chuveiro) + 01 (nº identificador). Refere-se ao medidor do chuveiro de nº 1 localizado no IFBA *campus* Salvador, bloco K, sala 02 do terceiro andar;

Descrição Descrição do medidor e outras observações tais como fabricante, tamanho, precisão etc

Localização Local onde está instalado o medidor

Unidade de medida Unidade de medida utilizada pelo medidor. Ex.: Km/h, L/min, m^3 , KW, KV, KW/L

3. Usuário clica no botão "Cadastrar"
4. O sistema faz validação dos dados e caso sejam válidos, os dados são persistidos no banco de dados
5. O sistema exibe mensagem "Medidor cadastrado com sucesso"
6. O caso de uso é encerrado

8.3 [UC03] Cadastrar elementos do sistema

Esta funcionalidade permite o usuário cadastrar elementos que representam componentes físicos da instalação hidráulica para facilitar na visualização de dados do mesmo. Exemplos: chuveiro, torneira, tanque, cano, condensadora, entre outros.

Pré-condições: O usuário deve ser administrador e estar logado no sistema.

Fluxo de eventos

1. Na tela inicial do Portal SMA, no menu superior há a opção "Cadastrar Elementos". O usuário deve selecionar tal opção.
2. Usuário deve obrigatoriamente preencher todos os campos para efetuar cadastro:
 - Sigla** Sigla de até 4 caracteres que refere-se ao elemento ressaltando que a mesma fará parte do nome do TAG. Exemplo: CH para chuveiro, T para torneira;
 - Nome** Nome propriamente dito do elemento; Exemplo: chuveiro, torneira, condensadora, descarga;
3. O sistema faz validação dos dados dos campos preenchidos e caso desejam válidos, esses são persistidos no banco de dados;
4. O sistema exibe mensagem "Elemento cadastrado com sucesso!"
5. O caso de uso é encerrado

8.4 [UC04] Consultar indicadores

Esta funcionalidade permite a visualização das informações de consumo de água do IFBA. Qualquer usuário tem acesso às informações sem a necessidade de cadastrar-se no sistema.

Fluxo de eventos

1. Na tela inicial do Portal SMA, no menu superior há a opção "Consultar Indicadores". O usuário deve selecionar tal opção.
2. O usuário pode escolher como opção de filtro os seguintes campos:
 - Medidor** Nome do medidor do qual deseja obter medidas
 - Data inicial e data final** Intervalo de tempo o qual deseja visualizar os valores de consumo de água.
 - Período** Período pré-determinado do dia. Ex.: madrugada, manhã, tarde e noite. Caso esta opção seja selecionada, os campos data inicial e data final estarão desabilitados.
 - Elemento** De que tipo de componente do sistema hidráulico deseja verificar consumo. Ex.: torneira, chuveiro, descarga, condensadora etc.
 - Localização** Local em que há medidores instalados. Caso esta opção seja selecionada o campo medidor estará desabilitado.

Caso o usuário selecione um medidor, o campo localização estará desabilitado assim como se uma localização for selecionada, o campo medidor estará desabilitado.
3. Um gráfico e tabela são gerados contendo as informações de consumo a partir dos filtros selecionados.
4. O caso de uso é encerrado