

Pluviômetro Digital

Rodrigo M. Druzian¹, Alessandro A. M. de Oliveira¹

¹ Sistemas de Informação - Centro Universitário Franciscano – Santa Maria, RS – Brazil

rodrigo.druzian@unifra.edu.br, alessandroandre@unifra.br

Abstract. *This paper presents the development of a Digital Gauge. One Arduino microcontroller board prototyping was used to take readings of the sensors connected to a simulation of a seesaw rain gauge. A system capable of readings was implemented, stores them in a memory card and with human intervention to generate a graph from an application to obtain data on rainfall.*

Resumo. *Este trabalho apresenta o desenvolvimento de um Pluviômetro Digital. Foi utilizado uma placa microcontroladora de prototipagem Arduino para fazer as leituras dos sensores ligados a uma simulação de um pluviômetro gangorra. Foi implementado um sistema capaz de fazer leituras, armazena-las em um cartão de memória e com a intervenção humana gerar um gráfico a partir de uma aplicação para obtenção de dados referentes às precipitações pluviais.*

1. Introdução

Os conhecimentos relativos ao tempo são um objetivo da meteorologia, estudados a partir de observações e experiências. Com a meteorologia foram criados instrumentos meteorológicos, sendo estes utilizados para adquirir dados meteorológicos, mas na sua grande maioria são mecânicos. Um destes instrumentos é o pluviômetro, que é usado para medir a quantidade de precipitação pluvial em milímetros, onde a precipitação pluvial é o nível de medição das águas das chuvas. [INMET, 2013]

O pluviômetro é um equipamento mecânico, na sua grande maioria, que não se utiliza de tecnologias atuais, ou seja, feito sob a ação humana ao ambiente, dispensando intervenção de um sistema informatizado. [INMET, 2013]

O pluviômetro mede a precipitação, mas ao juntarmos um equipamento eletrônico (Arduino), além das medidas, podem-se gerar dados precisos, sem uma forte intervenção humana.

O Arduino é uma plataforma eletrônica open-source (código aberto), baseada em hardware e software, ou seja, uma placa que pode ser programada através de um computador para executar diversas funções. [ARDUINO, 2013]

Neste trabalho, é apresentado a simulação de um pluviômetro tipo balança ligado a um equipamento eletrônico utilizando microprocessadores da plataforma Arduino juntamente com uma aplicação para melhor visualização dos dados. Além disso, são apresentados trabalhos relacionados ao assunto proposto.

2. Trabalhos Correlatos

Nesta seção, são abordados trabalhos relacionados com o escopo da presente proposta.

2.1. Sistema Automatizado de Aquisição de Dados Meteorológicos

O trabalho da Universidade Federal da Paraíba desenvolvido por Brito, Souza e Paz (1989), tem por objetivo automatizar três aparelhos meteorológicos através de um computador construído pelos mesmos para obterem dados precisos para pesquisas e monitoramento do lugar em questão. Desenvolveram sensores ligados ao computador onde podiam obter os dados de duas formas: leitura direta de cada sensor ou aquisição de dados automaticamente por períodos pré-estabelecidos sendo salvos em uma memória não volátil e depois de lidos em outro computador comum.

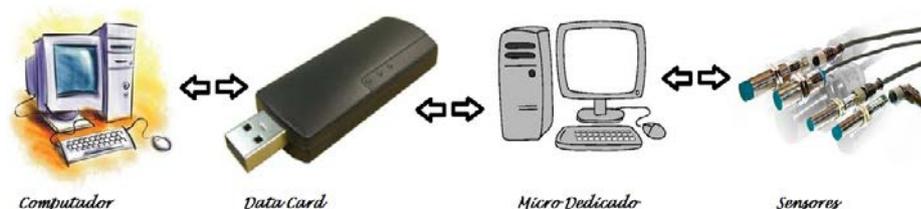


Figura 1 – Ambiente de desenvolvimento [adaptado de BRITO,SOUZA, PAZ, 1989]

Foi desenvolvida uma rotina de trabalho que consistia de: definição dos parâmetros a serem observados, fixação da data e horário a partir das quais os dados devem ser armazenados, ao final do período de aquisição transportar o cartão de dados para a central e processamento e arquivo de dados em discos flexíveis de microcomputadores. Conseguiram automatizar os instrumentos por eles ditos, diminuindo seu custo e difundido rapidamente na região.

2.2. Desenvolvimento de Plataforma para Monitoramento “Automatizado” de dados Termo Pluviométricos

Neste trabalho da Universidade do Estado Do Rio de Janeiro realizado por Ricardo Augusto Calheiros Miranda e Fernando Reizel Pereira tem por objetivo a concepção de uma plataforma de monitoramento de dados termo pluviométricos utilizando sensores inteligentes baseada em tecnologias de microcontroladores compactos, de custo e consumo reduzido de energia como uma nova alternativa para o monitoramento hidro ambiental e meteorológico. A configuração da estação termo pluviométrica se baseia em protótipo de pluviômetro do tipo balança. O esquema de coleta de monitoramento foi configurado para registrar o momento de basculhada, sendo a precipitação calculada individualmente para cada intervalo entre dois registros subsequentes. Os dados obtidos foram transformados em séries de pulsos registrados em intervalos de tempo fixos.

Os resultados obtidos na estação termo pluviométrica de baixo custo proposta apresentou acurácia e precisão quando foram Inter comparados com dados regularmente monitorados por estação meteorológica “automática”.

2.3. Comparação entre trabalhos

Os dois trabalhos relatam sobre um “pluviômetro digital”, porém há um custo muito elevado para os tipos de equipamentos utilizados. Aquém disto, os dois trabalhos precisam ser monitorados periodicamente, com certo tipo de intervenção humana para tratamento de dados para discos de microcomputadores ocupando muito espaço, ao qual se diferencia deste em questão por se tratar de um *hardware* livre, com custo baixo, sem ocupar muito espaço em disco, podendo ser deixado por um longo período de tempo em determinado lugar onde haverá intervenção humana para retirar os dados e transportá-los para a aplicação.

3. Referencial Teórico

Esta seção apresenta conceitos e tecnologias relacionados à este trabalho, referenciando um instrumento meteorológico usado para precipitações pluviais juntamente com um equipamento eletrônico para ser automatizado.

3.1. Pluviômetro

O pluviômetro é um instrumento meteorológico utilizado para recolher e medir a quantidade de líquidos e ainda sólidos como chuva, granizo e neve. Como o equipamento mede a quantidade de chuva que precipita, seu estudo é elementar para a meteorologia em conjunto com o sistema de temperatura. [RAINGAUGE, 2013]

Quando há a precipitação, os meteorologistas utilizam o pluviômetro para coletar a água das chuvas. Depois de um determinado tempo de exposição, a água que está lá dentro é medida assim mostrando o volume de chuva que caiu. [RAINGAUGE, 2013]

3.1.1. Tipos de Pluviômetros

Existem vários tipos de medidores de chuva disponíveis no mercado, desde simples recipientes para coleta da chuva até outros sistemas mais avançados como, sensoriamento remoto, radar, satélites, etc. Uma das classificações deve-se aos seus princípios de funcionamento, podendo ser eletromecânicos, mecânicos, piezoelétricos, ópticos, entre outros:

Pluviômetro Ordinária (medidor de chuva padrão): é o medidor mais comum, que consiste em um lugar acima de um coletor de funil que conduz ao cilindro de medição, onde se armazena a água das chuvas.

Pluviômetro Tipping (medidor automatizado de chuva): conhecido como pluviômetro de tombamento, usa pequenos baldes contendo uma quantidade específica de água das chuvas e contra balanceada por pequenos pesos, onde a água contida era despejada em um fita de papel, onde os furos nesta fita mediam a precipitação.

Pluviômetro de Ville de Paris : um dos pluviômetros mais simples, usado para captar e acumular a água da chuva para posterior medição com provetas graduadas.

Pluviômetro por peso : é um pluviômetro utilizado para pesar a água no interior do recipiente. O recipiente fica em cima de uma grelha que pesa junto com a água da chuva para dentro de forma contínua, com as escalas podendo ser ajustadas.

Pluviômetro de Bâscula: se utiliza de dois baldes, onde um balde bate, o outro sobe imediatamente e este último esvazia, assim vice-versa, por um determinado período de tempo. A cada batida, um sinal é enviado a um gravador, que registra o momento em que o balde bate, assim fazendo uma contagem e sendo visto quando um operador humano irá reiniciá-lo ou removê-lo.

3.2. Sensores

São dispositivos que alteram seu comportamento sobre uma grandeza física, ou seja, alteram seu tipo de energia para outro. [RUDEA, 2005]

Existem dois tipos, sensores analógicos e digitais.

Os sensores analógicos são aqueles que emitem sinais analógicos. São sinais limitados por dois valores de tensão, entretanto entre este limite, eles podem assumir intermediários infinitos, onde cada tensão medida terá um valor correspondente. [PATSKO, 2013]

Os sensores digitais só podem produzir dois valores binários ao seu sinal de saída, sendo zero (*low*) ou um (*high*), pois trabalham com níveis de tensão bem definidos. [PATSKO, 2013]

3.3. Microcontroladores

Os microprocessadores são máquinas que podem armazenar instruções lógicas, aritméticas e de tomada de decisão. [BASTOS, 2013]

Com o passar do tempo, os microprocessadores foram modificados (não todos) se tornando microcontroladores (circuitos integrados), ou seja, além de armazenarem instruções, podem ser programados para funções específicas. Internamente, microcontroladores são constituídos de CPU, interruptores, memória RAM, EEPROM e portas de entrada e saída. [BASTOS, 2013]

3.4. Arduino

Arduino é um *hardware* livre, projetada com um microcontrolador de placa única idealizada no ano de 2005 na Itália. [BOEIRA, 2013]

O objetivo desta pequena placa é o desenvolvimento de programas e ferramentas com baixo custo, e de fácil utilização. Sua linguagem de programação é essencialmente C/C++. [MCROBERTS, 2011]

Como é uma plataforma livre, *open-source* e *open-hardwares*, não são necessários nenhuma permissão para o desenvolvimento desde que sejam cumpridos os padrões de licença. [MCROBERTS, 2011]

3.4.1. Estrutura do Algoritmo do Arduino

A estrutura básica de programação do Arduino é chamada de *Sketch*, que encapsula o bloco de funções principais: *setup* () e *loop* (). A função *setup* () é usada para inicialização da placa e do programa. Executada quando a placa é ligada ou resetada através do botão. Ainda nesta sessão é informado para o *hardware*, o que iremos utilizar.

A função *loop ()* funciona como à *main ()* da placa. O programa escrito dentro da *loop ()* é executado indefinidamente, ou seja, ao terminar a última linha de execução da função em questão, o programa reinicia a partir da primeira linha da *loop ()* e continua a execução até que a placa seja desligada ou o botão de reset seja pressionado.

3.5. MicroSD Shield

O arduino possui outras placas que se acoplam a placa original sendo chamadas de Shields. Estas Shields agregam diversas outras funcionalidades ao Arduino. Uma delas é a MicroSD Shield que é basicamente para o acoplamento de um cartão de memória para gravação de dados. [ARDUINO, 2013]

3.6. Cartão de Memória

Os cartões de memória possibilitam guardar o máximo possível de qualidade em diversos formatos, sem uma fonte de energia.[CARTÃO DE MEMÓRIA, 2010]

Os cartões de memória se dividem em tipos, um deles, e atualmente expandido mundialmente, é o SD (*Secure Digital*) que visa melhor desempenho na transferência de dados e compatibilidade com determinação de seguranças da *Secure Digital Music Initiative* (SDMI). [ALECRIM, 2011]

4. Metodologia

A Metodologia utilizada para este trabalho foi a metodologia de desenvolvimento de *software* ICONIX. O ICONIX é considerado uma metodologia ágil, prático e simples. Desenvolvido nos anos 90 por *Doug Rosenberg e Kendall Scott*. Por se tratar de uma metodologia flexível e aberta, se em alguma de suas fases precisarem algum tipo de complemento, não há restrições para o uso de diversos recursos.

Este tipo de Metodologia se divide basicamente em quatro etapas:

- 1- Análise de requisitos
- 2- Análise e Projeto preliminar
- 3- Projeto
- 4- Implementação

Abaixo será descrito cada etapa com maiores detalhes.

4.1. Análise de Requisitos

Refere-se à primeira etapa da Metodologia ICONIX. Nesta parte é necessário levantar e descrever os requisitos do sistema, apresentar o caso de uso e seus respectivos atores e se possível apresentar um protótipo de interface ou um diagrama de modelo de domínio.

4.1.1. Descrição do Sistema

O projeto coleta dados da simulação de um pluviômetro ligado a sensores que serão interpretados por um microcontrolador de prototipagem Arduino. O sistema conta com uma gangorra(simulando um pluviômetro tipo báscula) onde em cada lado da báscula é ligado um sensor, totalizando em dois sensores.

Os sensores são ligados a um microcontrolador programado por prototipagem Arduino. O microcontrolador faz a leitura dos dados recolhidos. Os dados recolhidos são salvos em um cartão de memória, ao se retirar o cartão de memória e importa-los para uma aplicação que gera o gráfico.

Requisitos:

Dois botões (sensores de pulso), uma placa Arduino, um micros Shield, um cartão de memória. A gangorra foi utilizada para ir da esquerda para direita, ou vice-versa. Os sensores coletaram os dados de cada “batida”, onde a batida é quando a gangorra vai para esquerda ou para direita, em cada lado é uma “batida”. O Arduino recebe estes dados e envia os dados ao cartão de memória. O cartão de memória salva os dados. Os dados do cartão de memória são exportados para a aplicação. A aplicação gera o gráfico com a precipitação.

4.1.2. Diagrama de Caso de Uso

O diagrama de caso de uso tem por objetivo auxiliar a comunicação entre o analista e o cliente, descrevendo um cenário que demonstra as funcionalidades do sistema do ponto de vista do usuário. Abaixo se encontra o diagrama de Caso de Uso para Pluviômetro Digital desenvolvido:

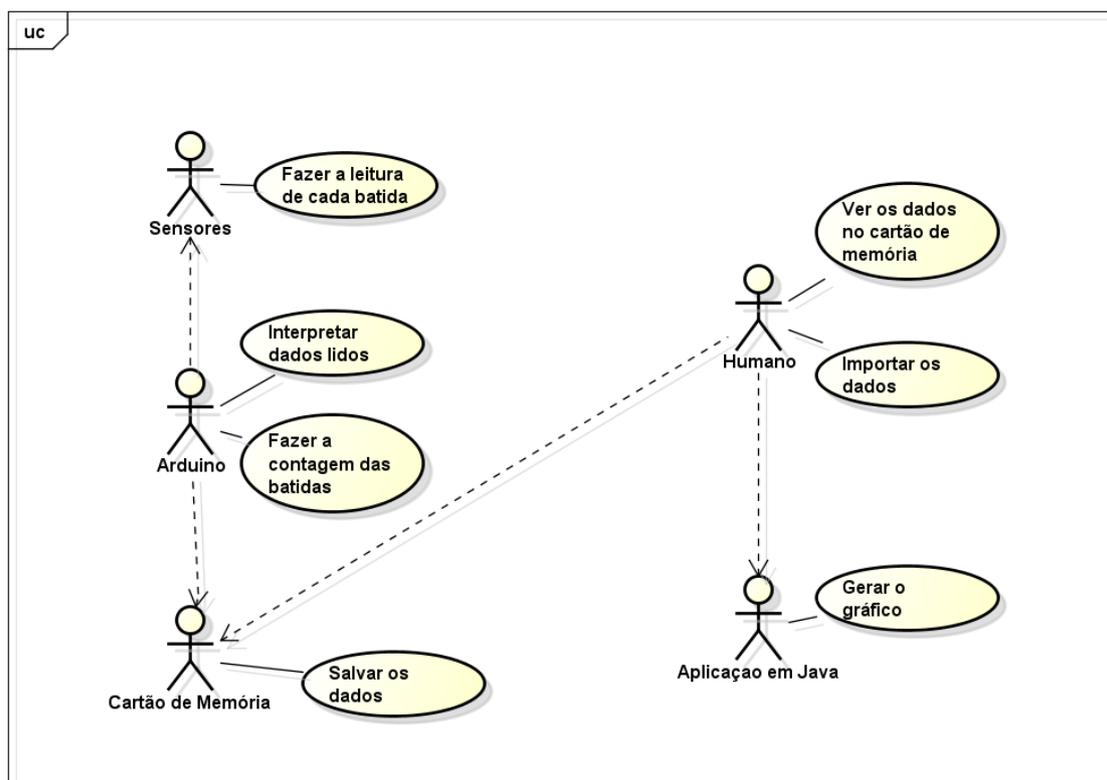


Figura 2 – Diagrama de Caso de Uso

Na seção 4.2.1 será descrito o caso de uso de forma integral.

4.1.3. Protótipo

Quando há o início da chuva, o pluviômetro começa a coletar sua precipitação. A balsa faz a medição do volume de chuva, os sensores do Arduino devem estar ligados a cada extremidade da balsa.



Figura 3 – Protótipo

Para o protótipo do Arduino para a coleta dos dados foram utilizados oito fios, dois botões, dois resistores, uma placa Arduino e uma Shield onde deve-se ter uma entrada para o cartão de memória.

- Dois fios ligados a uma extremidade dos resistores, ligados ao negativo da protoboard.
- Dois fios ligados a uma extremidade dos botões, ligados ao positivo da protoboard.
- Um fio interligado no negativo da protoboard, ligado ao arduino, no GND(terra).
- Um fio interligado no positivo da protoboard, ligado ao arduino, no 5V(cinco volts).
- Um fio ligado à outra extremidade do botão, ligado a porta digital número 6.
- Um fio ligado à outra extremidade do botão, ligado a porta digital número 3.

4.2. Análise e Projeto Preliminar

Refere-se à segunda etapa da Metodologia ICONIX. Nesta parte será apresentado a descrição dos casos de uso, os fluxos de ações do sistema e apresentar o diagrama de classes.

4.2.1 Descrição dos Casos de Uso

Fazer a leitura de cada batida	
Sumário	Os sensores fazem a leitura dos valores
Ator Primário	Sensores
Fluxo Principal	1. Os sensores fazem a leitura dos valores 2. Os sensores enviam os dados ao Arduino
Interpretar dados lidos	
Sumário	O arduino interpreta os dados lidos pelos sensores
Ator Primário	Arduino

Fluxo Principal	1. O Arduino interpreta os dados lidos pelos sensores
Fazer a contagem das Batidas	
Sumário	O Arduino faz a contagem dos toques
Ator Primário	Arduino
Fluxo Principal	1. O Arduino faz a contagem dos toques
Salvar os dados	
Sumário	O cartão de memória guarda os dados gerados pelo Arduino
Ator Primário	Cartão de Memória
Fluxo Principal	1.O Arduino envia os dados ao cartão de memória 2.O cartão de memória guarda os dados enviados pelo Arduino
Ver os dados no cartão de memória	
Sumário	O humano verifica se os dados estão certos no cartão de memória
Ator Primário	Humano
Fluxo Principal	1.O humano verifica os dados
Importar os dados	
Sumário	O humano importa os dados para a aplicação em java
Ator Primário	Humano
Fluxo Principal	1.O humano importa os dados para a aplicação em Java
Gerar gráfico	
Sumário	A aplicação em Java gera o gráfico dos dados coletados
Ator Primário	Java
Fluxo Principal	1.O Java gera o gráfico

4.2.2. Diagrama de Atividades

O diagrama de atividades apresenta os fluxos de ações do sistema.

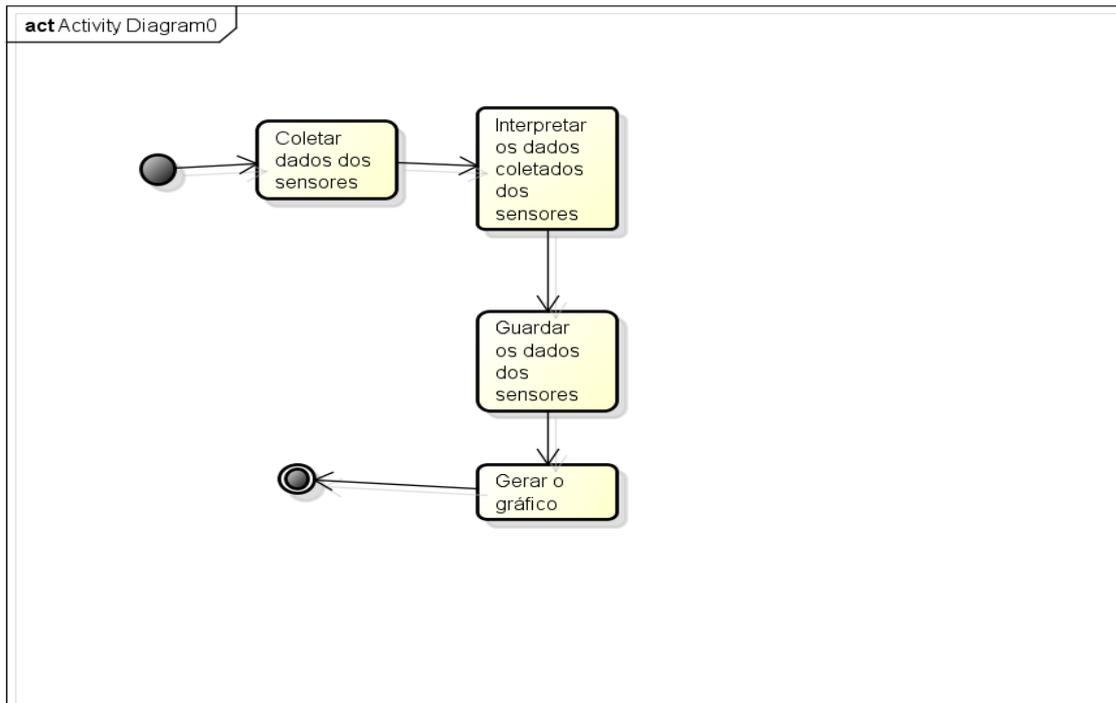


Figura 4 – Diagrama de Atividade

4.2.3. Diagrama de Classe

O Diagrama de Classe é considerado a evolução do Diagrama de Modelo de Domínio, mostrando detalhes específicos.

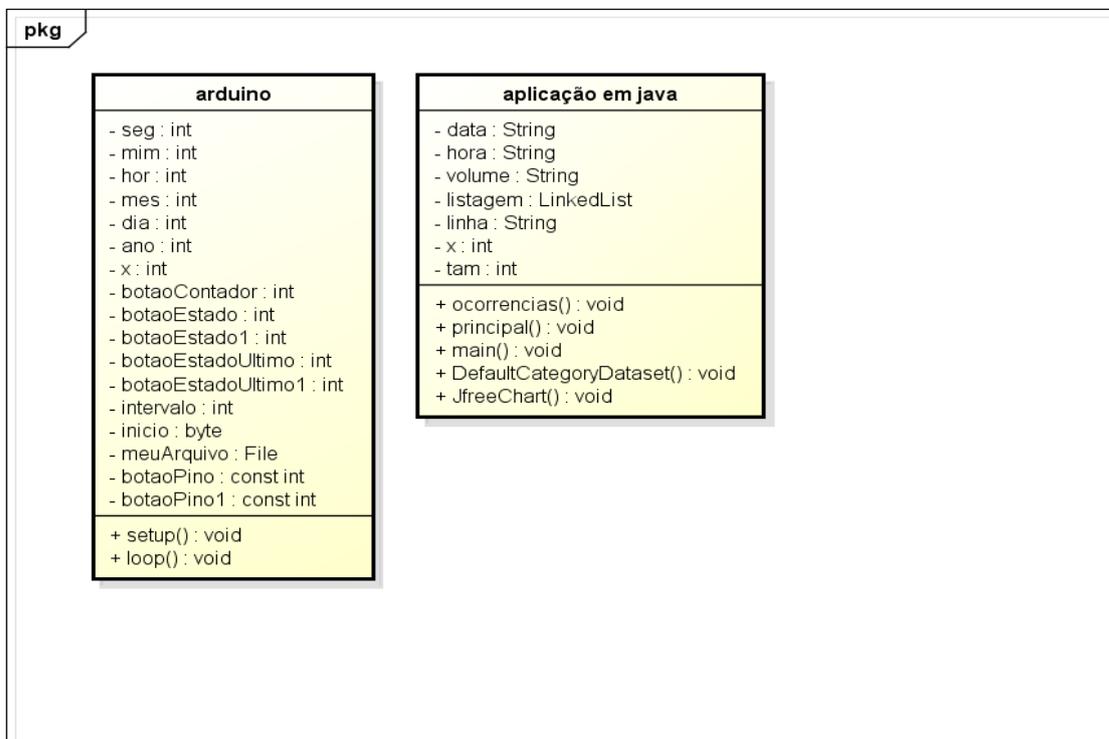


Figura 5 – Diagrama de Classe

4.3. Projeto

Refere-se à terceira etapa da Metodologia ICONIX. Nesta parte é apresentado o diagrama de sequência para especificar os componentes integrados do sistema.

4.3.1. Diagrama de Sequência

O Diagrama de Sequência apresenta as iterações do sistema.

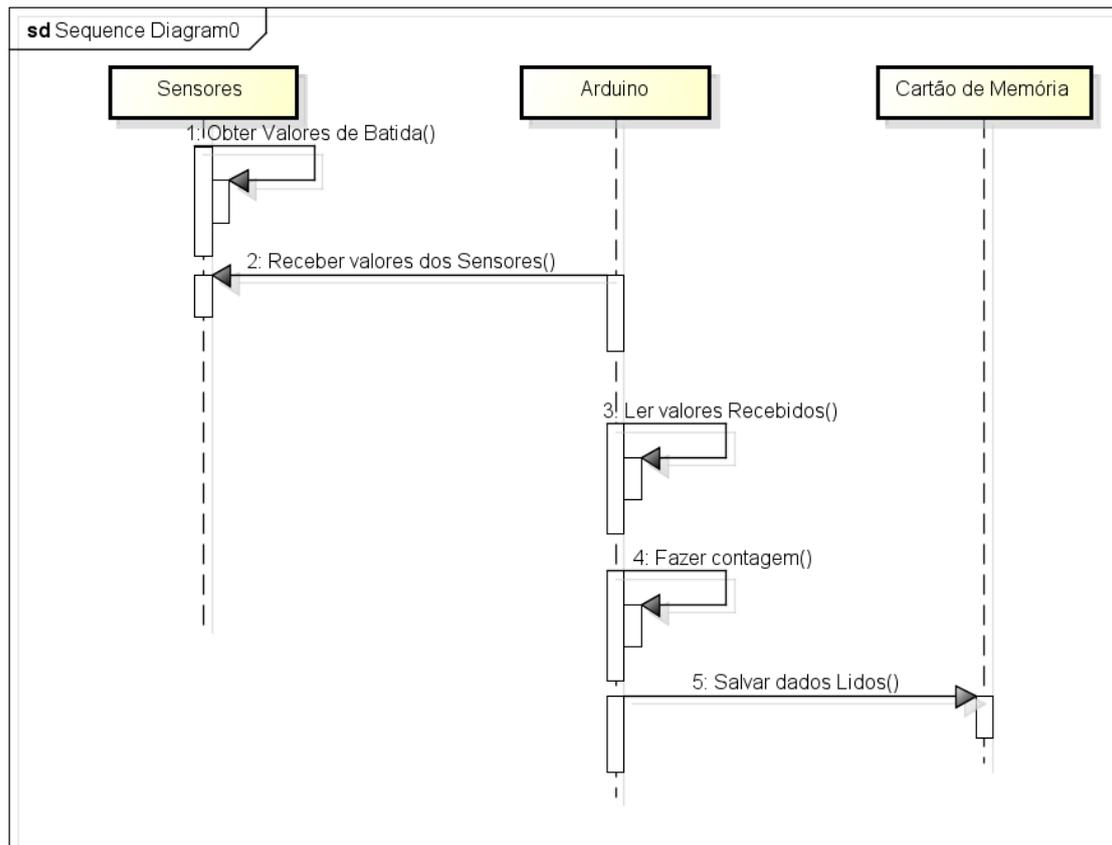


Figura 6 – Diagrama de Sequencia (1ª parte)

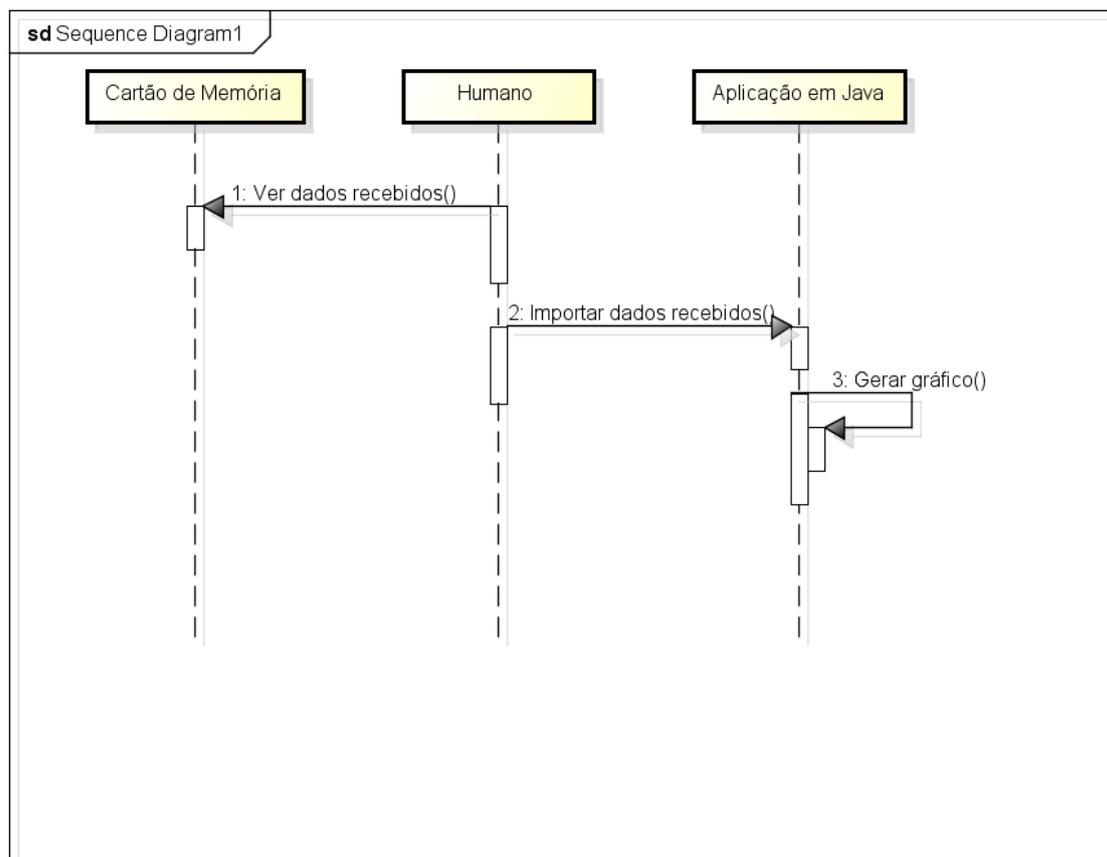


Figura 7 – Diagrama de Sequencia (2ª parte)

4.4. Implementação

Nesta parte da metodologia ICONIX são feitos os testes e o programa em si (código), salvos em algum tipo de disco rígido.

4.4.1. Testes

Foi delimitado um tempo menor para os testes, exatamente dez segundos. Ao ser pressionado um dos botões, o Arduino lia este dado e fazia a contagem aumentar para +1, diversificando de um botão para o outro fazendo a contagem aumentar em +1 cada vez que fosse pressionado um dos botões. Ao finalizar o tempo de 10 segundos, era mostrada no próprio serial, a data, seguida do horário mais a contagem de quantas vezes foram pressionados os botões.

Logo após o tempo delimitado passar, o contador era zerado e o tempo de dez segundos passado novamente.

Data	Hora	Volume
11/4/2014	8:6:10	2mmÂ³
11/4/2014	8:6:20	0mmÂ³
11/4/2014	8:6:30	1mmÂ³

Figura 8 – Serial Arduino mostrando zero

Feito estes testes, foi mandado salvar os dados no cartão de memória, mostrando a data, o horário e a quantidade de volume (contador).

```

Salvando dados
Mostrando dados dentro do cartao de memÃ³ria
prep.txt:
Data      Hora      Volume
11/4/2014 8:6:10    2mmÂ³
Salvando dados
Mostrando dados dentro do cartao de memÃ³ria
prep.txt:
Data      Hora      Volume
11/4/2014 8:6:10    2mmÂ³
11/4/2014 8:6:20    0mmÂ³
Salvando dados
Mostrando dados dentro do cartao de memÃ³ria
prep.txt:
Data      Hora      Volume
11/4/2014 8:6:10    2mmÂ³
11/4/2014 8:6:20    0mmÂ³
11/4/2014 8:6:30    1mmÂ³

```

Figura 9 – Serial Arduino mostrando os dados salvos no cartão

Após todos os testes, foi mostrado como os dados ficariam no cartão de memória.

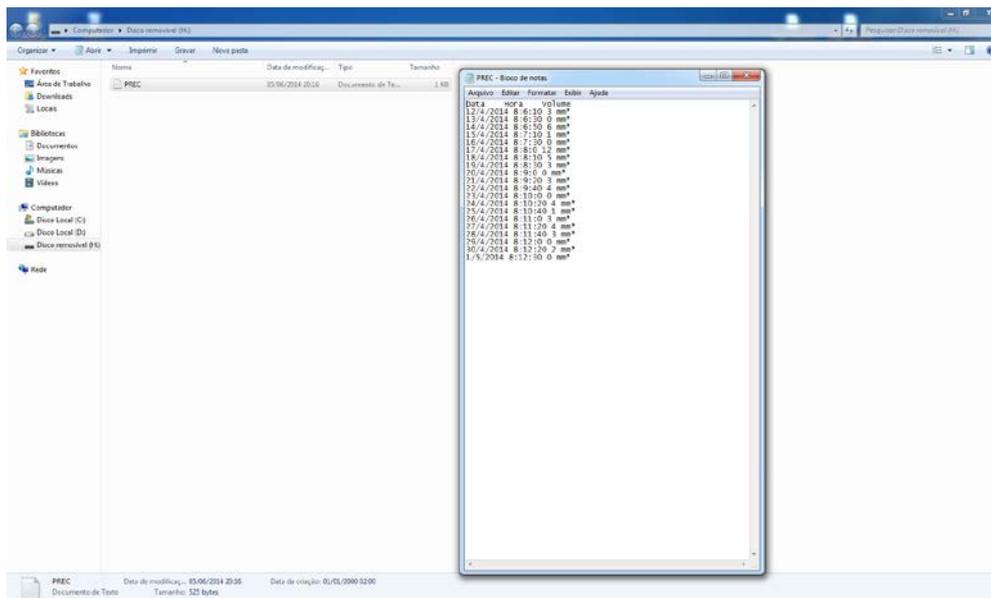


Figura 10 – Dados no cartão de memória.

Os dados que estão no cartão de memória são exportados para um aplicativo em Java que além de pegar estes dados, gera um gráfico mostrando a precipitação com sua respectiva data.

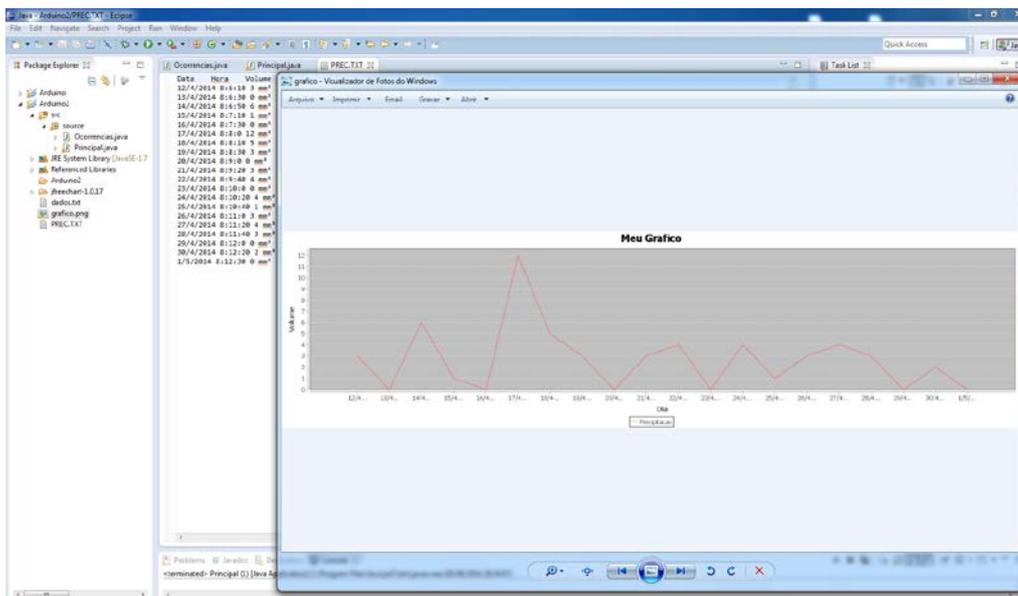


Figura 11 – Aplicação em Java

4.4.2. Pseudocódigos

Nesta subseção são mostrados partes do código do programa desenvolvido para a simulação de um pluviômetro.

4.4.2.1. Pseudocódigos Arduino

```
void loop() {
```

```

if (inicio) {
    seg = (seg + (intervalo / 1000));
    long tempoAnterior = millis();
    while (millis() - tempoAnterior < intervalo) {

```

Se existir inicio, ou seja, se o cartão de memória estiver conectado então começa a rodar o programa. Como é utilizado o *millis()*, deve-se dividir por 1000 para pegar somente segundos, assim incrementando. Enquanto o tempo passa em milissegundos e for menor que o intervalo fornecido(no caso 24 horas) ficará rodando o que vier abaixo do *while*.

```

        botaoEstado = digitalRead(botaoPino);
        botaoEstado1 = digitalRead(botaoPino1);
        if (botaoEstado != botaoEstadoUltimo) {
            if (botaoEstado == HIGH) {
                botaoContador++;
            }
        }
        if (botaoEstado1 != botaoEstadoUltimo1) {
            if (botaoEstado1 == HIGH) {
                botaoContador++;
            }
        }
        botaoEstadoUltimo = botaoEstado;
        botaoEstadoUltimo1 = botaoEstado1;

```

O *botaoEstado* e o *botaoEstado1* recebem a leitura pelas portas digitais de seus botões. Caso o estado do botão seja diferente do último estado e se o botão for pressionado, há a contagem de +1 para o contador, para qualquer um dos dois botões. Depois o *botaoEstadoUltimo* e o *botaoEstadoUltimo1* recebem os seus estados, ou seja, se foram pressionados ou não.

```

void setup() {
    inicio = 1;
    pinMode(botaoPino1, INPUT);
    pinMode(botaoPino, INPUT);
    pinMode(10, OUTPUT);
    Serial.begin(9600);
    if (!SD.begin(4)) {
        Serial.println("erro");
    }
}

```

```
    inicio = 0;
    return;
}
```

O programa passará somente uma vez por esta parte, onde é mandado os botões e a forma que eles são, se são de saída ou de entrada. Ainda há a verificação se o cartão de memória está presente, ele estando presente, o programa inicia e passa para a parte do *loop()*.

4.4.2.2. Pseudocódigo Aplicação em Java

```
Ocorrencias ocorrencia = new Ocorrencias();
    String arg[] = linha.split(" ");
    ocorrencia.setData(arg[0]);
    ocorrencia.setHora(arg[1]);
    ocorrencia.setVolume(arg[2]);
    principal.listagem.add(ocorrencia);
```

Nesta parte é estanciada a classe ocorrência. Cada espaço que há no arquivo .txt é mandado salvar em ocorrência, a data, a hora e o volume para após ir para uma lista com estes dados.

```
JFreeChart grafico = ChartFactory.createLineChart("Meu Grafico", "Dia",
    "Volume", ds, PlotOrientation.VERTICAL, true, true, false);
    OutputStream arquivo = new FileOutputStream("grafico.png");
    ChartUtilities.writeChartAsPNG(arquivo, grafico, 1700, 500);
```

Neste pseudocódigo é utilizado uma biblioteca chamada *JFreeChart* que cria o gráfico com o dia e o volume, gerando um arquivo de saída chamado gráfico.png e definindo o tamanho do gráfico para 1700x500.

5. Resultados Obtidos

Nesta seção serão demonstrados os resultados obtidos durante os testes, na execução do programa proposto.

5.1. Dados obtidos

Como demonstrado na implementação logo acima, o programa mostra os dados obtidos com precisão, ao botão ser pressionado conforme o vídeo feito, simulando uma gangorra. No serial, salvando os dados no cartão de memória, mostrando os dados dentro do cartão exportando-os para uma aplicação que gera um gráfico com a precipitação.

6. Conclusões

A ideia deste artigo consiste no estudo e desenvolvimento de um pluviômetro digital, usando um microcontrolador de prototipagem Arduino junto à simulação de um

pluviômetro para ter dados precisos e basicamente sem a intervenção humana. Feito de uma forma automática, sem gerar maiores custos e podendo ser usado em conjunto com outros instrumentos meteorológicos. Houveram dificuldades encontradas em relação à procura por material, principalmente em livros, de determinado assunto para o conhecimento básico para o desenvolvimento, mas o tema escolhido atendeu a expectativa, um projeto de fácil entendimento para leigos no assunto.

A metodologia escolhida foi satisfatória, pois além de ser uma metodologia de fácil entendimento, é bastante flexível podendo ser substituída por alguma outra coisa em alguma de suas etapas. A implantação de um sistema utilizando um microcontrolador de prototipagem Arduino é viável, além do qual é um custo relativamente baixo para suprir a demanda, visto que a tecnologia da informação está em crescimento constante ao redor de todo o mundo.

7. Referências

- Alecrim, Emerson. (2013) Tudo sobre Cartões SD (Secure Digital),
<http://www.infowester.com/sd.php#historia>
- Arduino.(2013) Site Oficial do criador,
<http://www.arduino.cc>
- Bastos. Alex Vidigal Bastos. (2013) Microcontroladores,
<http://www.decom.ufop.br/alex/arquivos/bcc425/slides/Microcontroladores.pdf>
- Boeira, Marcelo.(2013) O que é Arduino?,
<http://blog.marceloboeira.com/arduino/o-que-e>
- Brito, Antonio A.S. Sousa, Antonio H. Paz, Marcelo de M.A.(2013) Sistema Automatizado de Aquisição de Dados Meteorológicos,
<http://www.cbmet.com/cbm-files/10-cf083a7224045d2a4d162578bbbb32f1.pdf>
- Cartão de Memória.(2013),
<http://www.cartadememoria.com/cartao-de-memoria>
- Inmet.(2013),
http://www.inmet.gov.br/html/informacoes/sobre_meteorologia/instrumentos
- McRoberts, Michael. (2011) Arduino básico,
Novatec: São Paulo, p. 456.
- Patsko, Luis F. (2013) Aplicações, funcionamento e utilização de sensores,
http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_elettronica_-_aplicacoes_e_funcionamento_de_sensores.pdf
- PlanetSeed.(2013),
http://www.planetseed.com/ptbr/posted_faq/5045435
- Raingauge. (2013),
<http://i.cs.hku.hk/~rain/RainGauge-English.php>
- Rudea, T. Cruzeiro. (2005) Controles automáticos do nível de poeira baseado em micro controlador, p. 53, Universidade Federal de Lavras. Lavras, MG
- Sondaterra.(2013),
<http://www.sondaterra.com/produto-367Pluvi%C3%B4metro%20por%20peso.xhtml>
- Tancredo, L. C. (2013) Metodologia de desenvolvimento de software com ICONIX,
<http://pt.scribd.com/doc/56081208/Artigo-sobre-ICONIX>
- Umapatent. (2013)Universidad de Malaga,
<http://umapatent.uma.es/es/patent/pluviometro4b5/>