



AUTOMATIZAÇÃO DA DOSAGEM DE ÁGUA NO PROCESSAMENTO TRADICIONAL DO AÇAÍ UTILIZANDO ARDUINO

Jacielle Ferreira Costa – eng.jacielle@gmail.com¹
Lucas dos Santos Conde - condelucas19@hotmail.com¹
Lucas Ferreira Tavares - lucasmaple5@hotmail.com¹
Luciano da Silva Gouveia - luciano.s.gouveia@hotmail.com¹
Vanilze Vaz Monteiro - vanilzevaz@ymail.com¹

¹Universidade Federal do Pará
Rua Augusto Corrêa, 1 - Guamá
66075-110, Belém – PA

Resumo: *Este projeto propõe o desenvolvimento da automatização do processamento tradicional do açaí utilizando a plataforma Arduino, de forma que possa facilitar o controle do fluxo de água através da implementação de um circuito eletrônico, utilizando um sensor e uma válvula de controle de fluxo de água. A utilização do Arduino é importante para o desenvolvimento acadêmico dos discentes e serve de incentivo para o desenvolvimento de novos projetos. A programação desenvolvida viabiliza ao usuário a escolha do tipo de açaí que ele deseja obter após o processamento, ou seja, a densidade do mesmo, popularmente conhecido como açaí fino, médio e grosso. Através da programação utilizada para desenvolver a interface gráfica, o discente é incentivado a trabalhar com esta ferramenta que se torna importante para o desenvolvimento de plataformas que interajam com o usuário.*

Palavras-chave: *Arduino, Automatização, Programação, Açaí.*

1 INTRODUÇÃO

O processamento do açaí para consumo imediato é uma atividade comum na região norte do Brasil, sendo diariamente realizada por vários “batedores” de forma tradicional, utilizando uma máquina, na qual o “batedor” adiciona água de acordo com a densidade de açaí que se deseja obter. Este projeto foi pensando visando a otimização deste processo através de um dosador de água controlado pelo Arduino.

Em um contexto em que a automação dos processos antes realizados por força humana tem tomado grande parte do cenário mundial, o desenvolvimento de ideias que possam ser aplicáveis a atividades tradicionais tem surgido de forma exponencial. Dessa forma, processos como acender uma lâmpada ou medir o consumo de água de uma residência se tornaram possíveis através da utilização de microcontroladores associados a sensores eletrônicos e à lógica de programação. Assim, aplicou-se a ideia de um dosador

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



Promoção





de água no processamento do açaí, buscando inserir tais conhecimentos no cotidiano de um fabricante desta fruta regional.

2 FUNDAMENTAÇÃO TEÓRICA

O açaí é o produto extraído do epicarpo e do mesocarpo, partes comestíveis do fruto do açaizeiro, após amolecimento obtido por processos tecnológicos adequados (EMBRAPA, 2008). Conforme legislação vigente, o açaí processado é classificado em:

- Açaí grosso ou especial (tipo A): a polpa extraída com a adição de água apresenta, após ser filtrada, mais de 14% de sólidos totais e a aparência é muito densa;
- Açaí médio ou regular (tipo B): a polpa extraída com a adição de água apresenta, após ser filtrada, de 11% a 14% de sólidos totais e tem a aparência densa;
- Açaí fino ou popular (tipo C): a polpa extraída com a adição de água apresenta, após ser filtrada, de 8% a 11% de sólidos totais e a aparência é pouco densa.

3 METODOLOGIA

Para a implementação do projeto foram utilizados os seguintes materiais:

- 1 Sensor de fluxo de água;
- 1 Válvula solenoide;
- 1 Arduino Uno;
- 1 Módulo relé 5V;
- 3 Roscas plásticas $\varnothing \frac{3}{4} \times \frac{1}{2}$;
- 1 Bucha plástica $\varnothing \frac{3}{4}$;
- 2 Mangueiras de $\varnothing \frac{3}{8}$;
- 1 *Protoboard*;
- *Jumpers*.

3.1 Implementação

O primeiro passo para a implementação do controlador do fluxo de água foi calibrar o sensor. Para este procedimento fez-se necessário utilizar um programa de teste para verificar a quantidade de pulsos (N) emitidos em cada condição estabelecida pelo projeto. Desta forma, pode-se obter a razão entre N e a quantidade de água (L) em litros, a qual dividida por 60 segundos, resultará em um fator K como mostrado na equação 2. O fator será utilizado para converter a quantidade de pulsos em litros por minuto, de acordo com as equações abaixo:

$$Q = \frac{N}{L} \quad (1)$$

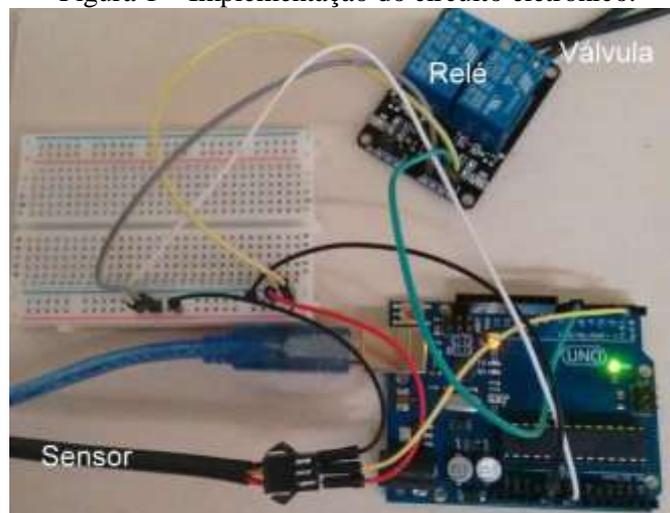
$$K = \frac{Q}{60s} \quad (2)$$

Este fator será utilizado na programação do controlador, permitindo que a vazão seja analisada por um resultado aproximado, tendo o desvio de no máximo 5%. É importante ressaltar que o fator K utilizado se deu pela média aritmética de todos os pulsos, para que as condições de vazão fossem satisfeitas.



Após a calibração do sensor foi feita a montagem do protótipo, acoplando-se a válvula solenoide em uma torneira. Em seguida, fez-se a ligação da válvula com o sensor através de uma mangueira. Posteriormente, realizou-se a implementação do circuito eletrônico, efetuando-se as ligações da seguinte forma: no relé, ligou-se a porta IN2 no pino 7 do Arduino, o GND diretamente na placa e o VCC em uma linha curto-circuitada na *proto-board* com o 5V do Arduino. Nos seus contatos, o comum (C) é ligado com a fase da corrente alternada, e um ramal da válvula é ligada ao normalmente aberto (NA) do relé. Seguidamente, foi realizada a ligação do sensor de tal forma que o Pino PULSO (D2) fosse conectado ao pino 02 do Arduino, o GND na placa e o VCC em um ponto curto-circuitado na *proto-board* com 5V do Arduino, conforme a Figura 1.

Figura 1 – Implementação do circuito eletrônico.



3.2 Programação

Primeiramente, foram definidas variáveis do tipo *float*: *vazão*- para armazenar o valor da vazão de água em $L=min$; *volume atual*- para medir o volume total detectado pelo sensor até o instante t ; *V_acaifino*, *V_acaimediao*, *V_acaigrosso*- que determinam o volume de água de deve ser despejado para cada respectiva densidade de açaí; e *mediavolume*- que define o volume de água que deve ser liberado em cada intervalo, sabendo que foram pré-estabelecidos três períodos para que o mesmo ocorra, considerando que o volume total para cada tipo de açaí foi dividido em três partes, conforme a Figura 2.

Figura 2 – Definição das variáveis.

```
float vazao;  
float volumeatual=0;  
int contaPulso;  
int i=0, j=0;  
float V_acaifino = 2.5;  
float V_acaimediao = 2.0;  
float V_acaigrosso = 1.5;  
byte valvula_rele=0;  
char densidade;  
float mediavolume;  
int a;
```



Em seguida, foram definidas as variáveis do tipo *int*: *contaPulso*- que armazena o valor da quantidade de pulsos que o sensor envia ao Arduino; *i*- que conta os segundos passados durante a execução do programa e que é zerado a cada minuto; *j*- que controla a repetição dos três intervalos; *a*- responsável por um *delay* de 2s. Definiu-se também, uma variável do tipo *char*, *densidade*- que recebe o tipo de açaí desejado pelo usuário, e uma variável do tipo *byte*, *valvula-rele*- que define o pino de entrada do relé o qual está conectado a válvula. O próximo passo foi definir o *void setup*, conforme a Figura 3.

Figura 3 – Código do *void setup*.

```
void setup() {  
  Serial.begin(9600);  
  void incpulso();  
  pinMode(valvula_rele, OUTPUT);  
  pinMode(2, INPUT);  
  digitalWrite(valvula_rele, HIGH);  
  attachInterrupt(0, incpulso, RISING);  
  Serial.println("\nDigite a densidade do acai:\n");  
}
```

O comando *Serial.begin* (9600) inicia a comunicação serial com o Arduino a uma taxa de 9600 *Bits/s*; *void incpulso* (), declara a função *incpulso*, responsável pelo incremento da variável *contaPulso*, conforme a Figura 4.

Figura 4 – Código do *void incpulso*.

```
void incpulso ()  
{  
  contaPulso++;  
}
```

O comando *pinMode(valvula_rele, OUTPUT)*, configura a porta do relé como porta de saída da placa; *pinMode(2, INPUT)*, define a porta 2 como entrada para o sensor. O comando *attachInterrupt(0, incpulso, RISING)* é uma rotina de interrupção, na qual configurou-se o parâmetro 0 para monitorar o pino 2. Dessa forma, toda vez que o Arduino detectar uma mudança do nível LOW para o nível HIGH nesse pino, a função *incpulso* será chamada.

Para a definição do *void loop*, utilizou-se, primeiramente, uma rotina de seleção *if* para identificar se há alguma informação na porta serial. Caso haja esta informação, a mesma será recebida pela variável *densidade*. A partir desse ponto, inicia-se o principal processo do programa, no qual de acordo com o que foi escolhido pelo usuário (ou seja, se foi selecionada a opção F - fino, M - médio ou G - grosso), irá se fazer o controle e a medição do volume de água que passará pelo sensor. Este processo inicia-se dividindo o volume de água total necessário para o processamento de cada tipo de açaí em três partes, para que o mesmo seja adicionado em três períodos separados por um intervalo de 30 segundos (para efeito de demonstração).

O controle disso é feito também por meio da estrutura de repetição *while* (com a variável de controle *j*), que realiza o mesmo processo três vezes. Analisando a parte interna do código, utiliza-se novamente a estrutura *while* dentro da qual se realiza a contagem dos pulsos enviados pelo sensor para o Arduino, a conversão dos mesmos em vazão (L/min) e a soma do volume passado pelo sensor até o instante atual. As funções



sei() e *cli()* habilitam e desabilitam a interrupção, respectivamente, para que assim, a cada segundo, possa ser detectada a quantidade de pulsos para o cálculo da vazão em L/min. Estas informações serão impressas na porta serial do Arduino e todo este procedimento é realizado até que o volume atual seja igual a média de volume calculada no início do *loop*. Assim que se passa o *delay* de 30s a variável *volumeatual* é resetada para que possa se fazer uma nova comparação ao até o fim da rotina *while*, ou seja, quando a variável de contagem *j* for igual a 3. Na figura 5, pode-se observar parte do código do *void loop*.

Figura 5 – Trecho do código do *void loop*.

```
void loop() {  
  j=1;  
  if(Serial.available() != 0){  
    densidade=Serial.read();  
    Serial.write(densidade);  
    if(densidade=='F'){  
      mediavolume=V_acaifino/3;  
      while(j<=3){  
        volumeatual=0;  
        while(volumeatual<mediavolume){  
          digitalWrite(valvula_rele,LOW);  
          contaPulso = 0;  
          sei();  
          delay (1000);  
          cli();  
        }  
      }  
    }  
  }  
}
```

O mesmo código se aplica para as outras opções (M ou G), mudando apenas o volume total o qual o hardware permitirá passar pelo circuito.

3.3 Processing

Visando uma demonstração gráfica do projeto, utilizou-se o *software Processing*, no qual é possível estabelecer uma interação com o projeto por meio da comunicação serial. Primeiramente cria-se as instâncias das imagens, importa-se a biblioteca serial do *Processing* e são declaradas as variáveis que serão utilizadas no programa.

Em seguida, o *void setup*, define-se o valor inicial das variáveis, carregam-se as imagens nas instâncias declaradas; define-se o tamanho da janela que será usada na interface e define-se o plano de fundo, conforme a Figura 6.



Figura 6 – Trecho do código do *void setup* implementado no *processing*.

```
d=0;
f=0;
size (800,600);
text01=createFont ("Arial",16,true);
String portName = Serial.list () [0];
myPort=new Serial (this, portName,9600);
botaoOnF=loadImage ("botaoOnF.png");
botaoOffF=loadImage ("botaoOffF.png");
botaoOnG=loadImage ("botaoOnG.png");
botaoOffG = loadImage ("botaoOffG.png");
botaoOnM = loadImage ("botaoOnM.png");
botaoOffM = loadImage ("botaoOffM.png");
back = loadImage ("backacai2.png");
background (back);
}
```

No *void draw*, iniciou-se com um comando que imprimiu na interface a mensagem inicial. Em seguida, colocou-se vários comandos que definiram as condições para cada imagem fosse carregada no momento certo. Na Figura 7, pode-se observar trecho do código.

Figura 7 – Trecho do código do *void draw*.

```
text ("Escolha o tipo de aço",width/2-145,100);

if (a==1&&c==1&&e==1)
{
for (i=height/2-76; i<=height/2+76; i++)
for (j=324; j<=476; j++)
set (j, i, branco);
for (i=height/2-76; i<=height/2+76; i++)
for (j=524; j<=676; j++)
set (j, i, branco);
for (i=height/2-76; i<=height/2+76; i++)
for (j=124; j<=276; j++)
set (j, i, branco);
image (botaoOffF, 124, height/2-76);
image (botaoOffG, 524, height/2-76);
}
```

Na Figura 8, pode-se observar a função *mouseClicked*, na qual foram definidas áreas, com as quais o operador irá interagir com a interface, através de cliques no mouse.

Figura 8 – Trecho do código do *mouseClicked*.

```
void mouseClicked () {
if (c==1&&e==1&&a==1) {
if (mouseX>=124&&mouseX<=276&&mouseY>=
height/2-76&&mouseY<=height/2+76)
{
myPort.write ('F');
}
```



Essas áreas correspondem às imagens dos botões da interface, também sendo definidas as condições para que essas áreas se tornassem interativas.

3.4 Palestra Motivacional

Para expor o projeto para a comunidade acadêmica foi promovida uma palestra para discentes do segundo bloco do curso de Engenharia Elétrica da Universidade Federal do Pará, contando com a presença de 20 discentes. No decorrer da palestra foram explicados os *softwares* utilizados para o desenvolvimento do projeto.

4 RESULTADOS E DISCUSSÕES

Após todos os testes realizados, chegou-se na implementação final do projeto que pode ser visualizada na Figura 9.

Figura 9 – Implementação final do projeto.

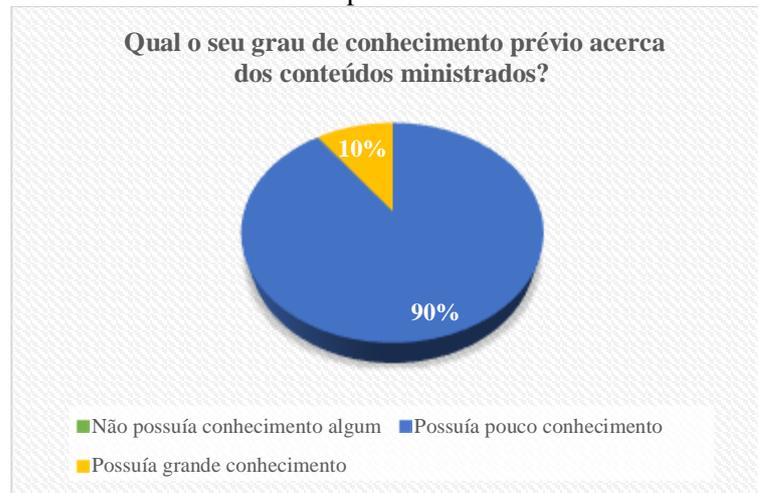


Como um dos objetivos do projeto se enquadra em divulgar para a comunidade acadêmica as possibilidades existentes para o desenvolvimento de projetos utilizando o Arduino, foi promovida uma palestra de divulgação do projeto. Ao final da apresentação os discentes presentes responderam a um questionário com perguntas relacionadas à apresentação, sua importância, o nível de seus conhecimentos prévios acerca do assunto, se a palestra despertou interesse dos mesmos para criar novos projetos, entre outros questionamentos. Dentre as perguntas destacam-se os gráficos com as respostas das questões mais relevantes.

Quando os discentes foram questionados quanto ao conhecimento/contato prévio com as linguagens de programação utilizada e com a plataforma Arduino, constatou-se que uma boa parte dos alunos teve esse contato, devido a este microcontrolador ser bastante utilizado nos cursos de engenharia elétrica, porém ainda assim estes possuíam pouco conhecimento sobre a plataforma e suas funcionalidades.

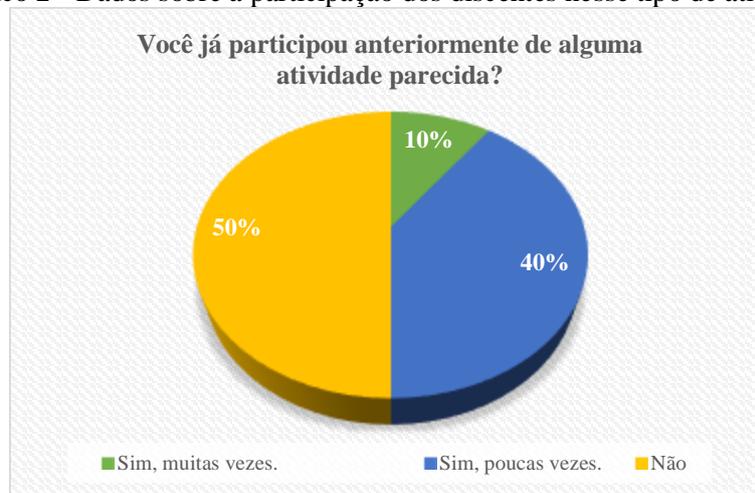


Gráfico 1 – Dado sobre o conhecimento da plataforma Arduino e as linguagens de programação antes da palestra.



Ao serem questionados a respeito da participação em alguma atividade como a palestra a promovida. Como mostra o Gráfico 2, 40% deles já haviam participado de alguma atividade parecida. Dessa forma, é importante que se promovam atividades que despertem aos discentes o interesse em utilizar ferramentas como o Arduino e o *processing* para o desenvolvimento de projetos.

Gráfico 2 - Dados sobre a participação dos discentes nesse tipo de atividade.



Os discentes também foram consultados com relação a interface gráfica criada no *Processing* e sua facilidade de entendimento desta pelo usuário. Com base na resposta dos discentes pode-se constatar que a plataforma é de fácil entendimento ao usuário como mostra o Gráfico 3, facilitando o entendimento dos discentes acerca do conteúdo mostrado pelo palestrante sobre a plataforma.

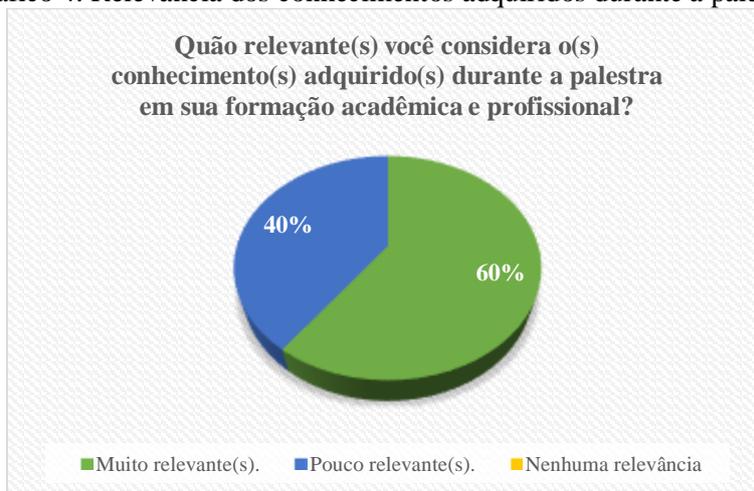


Gráfico 3: Facilidade de entendimento da interface gráfica do site pelo usuário.



Também foram questionados sobre a relevância do conhecimento adquirido durante a palestra para o processo de construção da formação acadêmica e profissional. Os resultados deste questionamento estão apresentados no Gráfico 4.

Gráfico 4: Relevância dos conhecimentos adquiridos durante a palestra.



Pode-se observar que 60% dos discentes que participaram da palestra, consideram relevantes os conhecimentos adquiridos durante a palestra.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento deste projeto utilizando Arduino permitiu benefícios consideráveis, principalmente quanto a praticidade da produção do açaí, no qual a adição de água se dá automaticamente. Além disso, é imprescindível mencionar que o uso do sensor proporcionou a racionalização dos recursos hídricos, como também a higienização no processo, haja vista que a água é adicionada sem ter contato com o “batedor”. No que tange os fatores negativos, tem-se a calibração do sensor, a qual fez-se necessária para



determinar o fator k , que varia de acordo com a pressão e pureza da água, bem como o número de pulsos por litro.

Quanto ao ensino acadêmico, pode-se considerar que o projeto desenvolvido foi de grande relevância, visto que no decorrer da palestra promovida houve um considerável interesse dos discentes e até mesmo sugestões de melhorias para o projeto. Esse aspecto pode ilustrar a forma como projeto instiga o interesse por desenvolver novos projetos utilizando o Arduino, sendo que é importante para o desenvolvimento acadêmico dos discentes. Considerando os resultados adquiridos durante a pesquisa, foi observado que é importante tornar, a interface gráfica de acesso do usuário, mais intuitiva, visto que uma certa porcentagem demonstrou dificuldades na utilização do mesmo. É importante destacar que a maioria dos discentes que participara da pesquisa possuem conhecimento prévio do assunto abordado, o que torna a palestra muito mais interessante e participativa.

REFERÊNCIAS BIBLIOGRÁFICAS

COHEN, K. O., ALVES, S. M. *Sistemas de Produção*, 4, 2a ed. ISSN 1809-4325 Versão Eletrônica, Dezembro, 2006.

Laboratório de Garagem, *Tutorial: Como criar sua própria interface gráfica para seus projetos utilizando Processing e Garagem*, labdegaragem.com . Data de acesso: 28 de janeiro de 2016.

Laboratório de Garagem, *Tutorial: Como utilizar um sensor de fluxo de água - G 3/4 com Arduino*, labdegaragem.com. Data de acesso: 06 de janeiro de 2016.

McRoberts, M., *Arduino básico*, 1a ed. [tradução Rafael Zanolli], São Paulo: Novatec, 2011.

AUTOMATIZATION OF WATER DOSAGE IN TRADITIONAL AÇAÍ PROCESSING USING ARDUINO

Abstract: *This project proposes the development of the automation of the traditional açai processing using the Arduino platform, so that it can facilitate the control of the water flow through the implementation of an electronic circuit, using a sensor and a water flow control valve. The use of Arduino is important for the academic development of students and serves as an incentive for the development of new projects. The programming developed enables the user to choose the type of açai that he wants to obtain after processing, that is, the density of the same, popularly known as thin, medium and coarse açai. Through the programming used to develop the graphical interface, the student is encouraged to work with this tool that becomes important for the development of platforms that interact with the user.*

Key-words: *Arduino, Automation, Programming, Açai.*

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



Promoção



Associação Brasileira de Educação em Engenharia