

Universidade Presbiteriana Mackenzie

Projeto Objetos Inteligentes Conectados

Automatização de produção de cerveja artesanal

Alunos:

Gustavo Trindade de Avila - 31515649

Joao Ailton Junior da Silva

Diego Muniz Sobrinho

Turma: 06J

A ideia inicial do projeto surgiu pela paixão a cerveja e enxergamos uma necessidade no mercado de em relação a produtores independentes. Hoje a cerveja é um dos produtos mais consumidos pelos jovens, por este motivo resolvemos empreender e criar um projeto onde podíamos aplicar o conhecimento obtidos em aula em algo que possamos usar no dia-a-dia.

O processo de produção de cerveja artesanal é composto por 7 passos (moagem, mosturação, recirculação, fervura, resfriamento, fermentação e maturação) o projeto busca automatizar e melhorar esses processos já que muitos deles dependem de extremo cuidado e atenção em pontos como por exemplo a temperatura.

Levando em consideração os pontos citados acima utilizamos em nosso projeto um termômetro DS18B20 para que o produtor da cerveja possa controlar a temperatura durante todo o processo, e dois motores DC, um deles ira controlar a derrubada do malte sobre a água fervendo e enquanto o malte é despejado sobre a agua o outro motor atua para já ir misturando esse malte e formando o mosto. Com o mosto formado o motor DC continua atuando sobre o mosto e fazendo o processo chamado de brasagem ou mosturação, que é o processo responsável por viabilizar a atuação enzimática que deve, dentre outras funções, converter o amido dos grãos em açúcares fermentáveis (processo chamado de maltose) e/ou não fermentáveis (dextrinas).

O termômetro é de extrema importância em todo o processo, pois durante todo o processo a temperatura tem uma função diferente sobre a cerveja produzida, como por exemplo, ainda durante a mosturação, com o controle de temperatura é possível ter um controle em processos como o de parada na acidificação, que ocorre com a temperatura entre 40°C e 55°C, ou o repouso proteico, com a temperatura entre 50°C e 55°C. A temperatura influencia também no tipo da cerveja, como por exemplo cervejas do estilo Ale exigem temperatura entre 17°C e 24°C durante o período de maturação enquanto a Lager devem ficar com a temperatura entre 9°C e 14°C, o processo maturação é o processo de armazenamento da cerveja em determinada temperatura para que ocorra alterações químicas que auxiliam a clarificação e melhoram o aroma e o sabor da cerveja.

O projeto consiste em automatizar o processo produção de cerveja artesanal através de um aplicativo chamado Blynk, as funcionalidades do aplicativo são:

- Mexer Mosto– aciona um dos motores DC para mexer a pá no balde.
- Liberar Malte – aciona outro motor DC que gira a broca para liberar o malte.
- Verificar Temperatura – apresenta a temperatura identificada pelo sensor de temperatura DS18B20.

O aplicativo irá fazer comunicação com o Arduino UNO para poder executar os comandos listados acima.

Objetivos:

Automatizar a produção de cerveja artesanal, permitindo o produtor fazer a cerveja de uma maneira mais rápida, menos cansativa, podendo produzir ainda mais cerveja de uma maneira inovadora.

Problema:

- Descrição do problema:

Atualmente o processo de produção de cerveja artesanal é um processo que leva muito tempo para ser concluído além de possuir algumas partes do processo que demandam uma segunda pessoa para auxiliar na produção, e também em alguns momentos é processo cansativo por exigir do produtor ficar mexendo constantemente o malte.

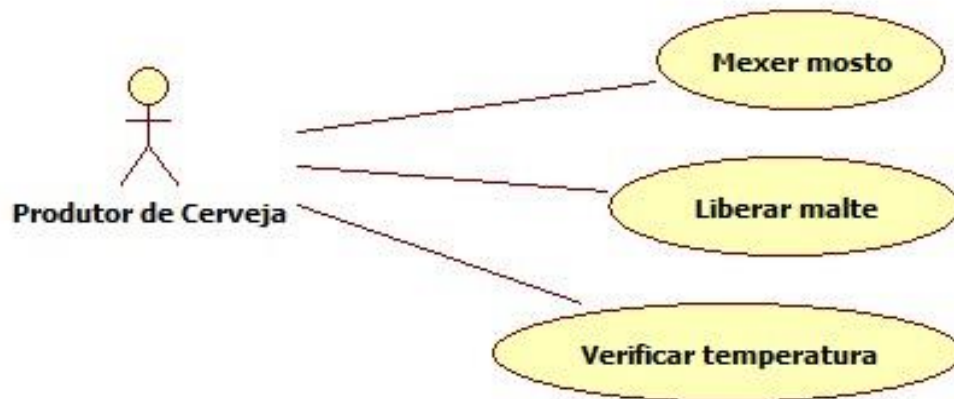
- Quem é afetado pelo problema:

Produtores de cerveja artesanal.

- Benefícios de uma boa solução:

Agilizar o processo de produção de cerveja, torna-lo mais simples e sem a necessidade de uma segunda pessoa além do produtor, além de fazer o processo menos cansativo, já que o produtor não terá necessidade de ficar mexendo o malte, atividade que será exercida por um dos motores DC.

Diagrama de Casos de Uso:



Plataformas de Desenvolvimento: **Fritzing e Arduino IDE**

Imagem Esquemático

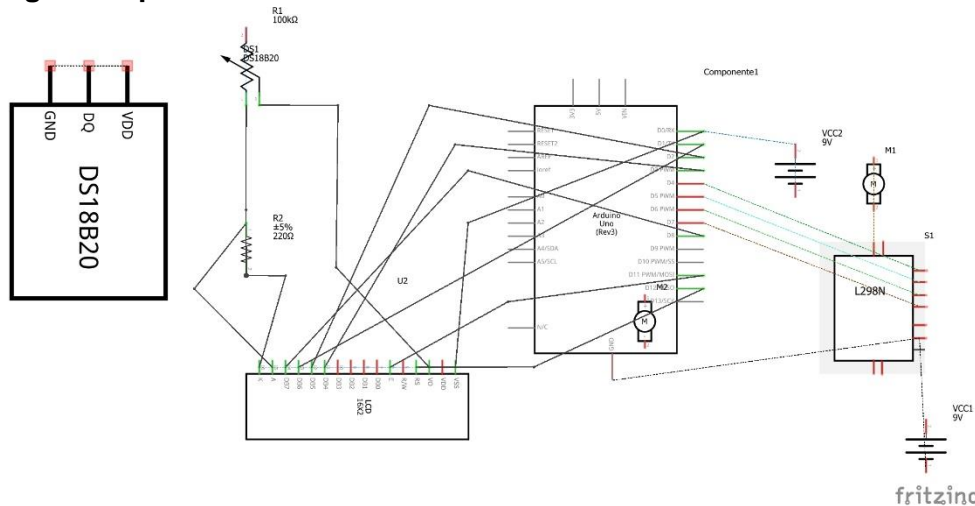
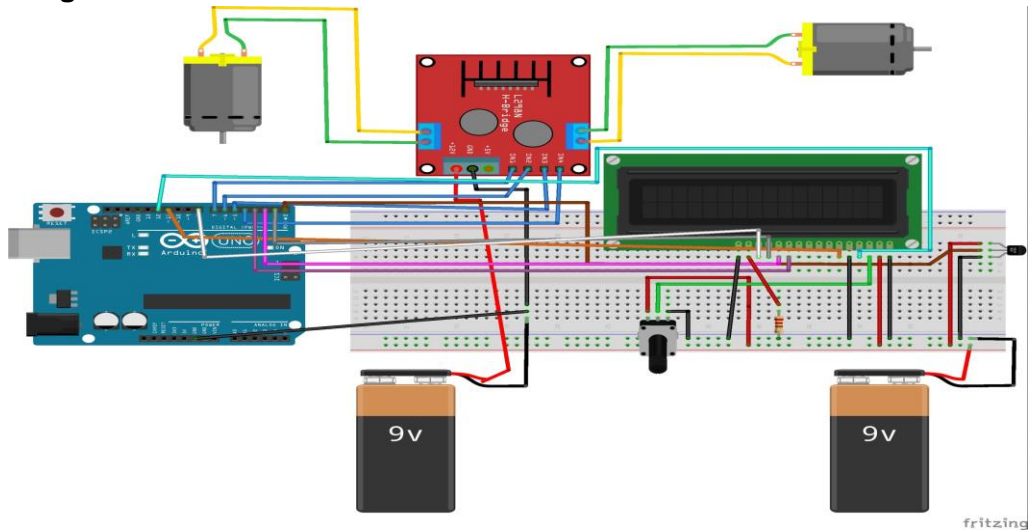
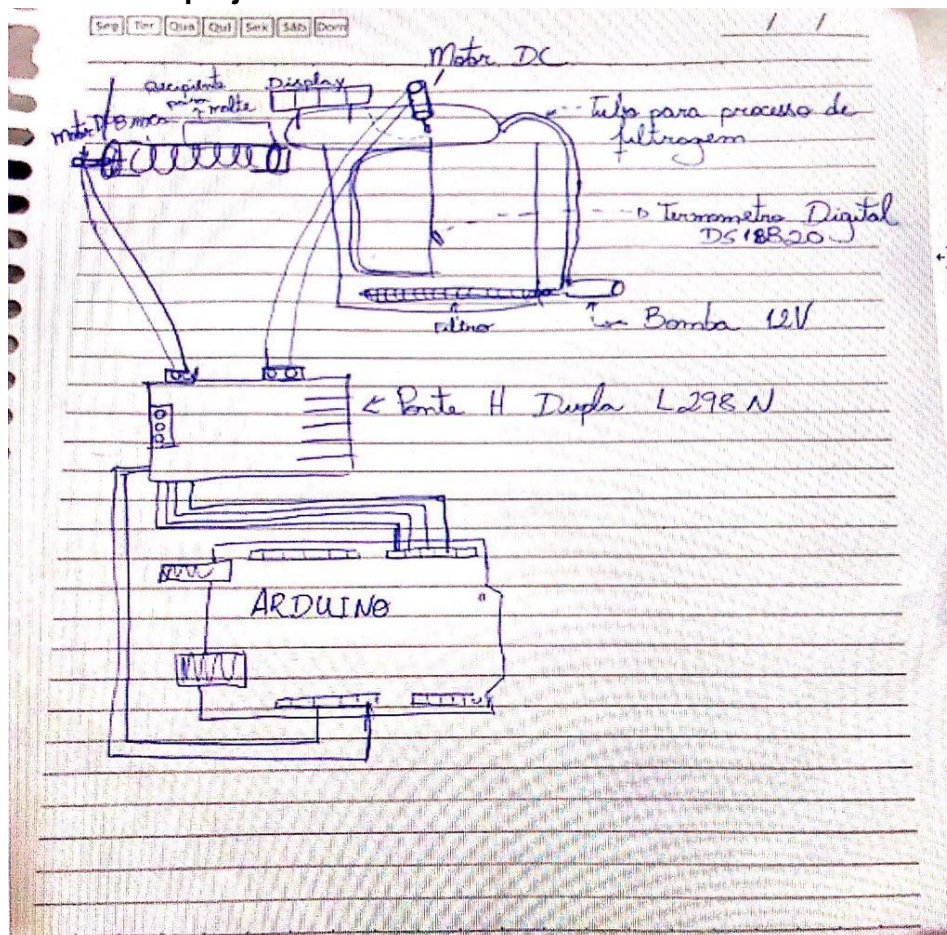


Imagem Protoboard.



Desenho do projeto:



Materiais utilizados no projeto:

Sensores:

1 Termômetro Digital DS18B20

Atuadores:

2x Motor DC

Displays:

1 Display LCD

Arduino:

1 Arduíno UNO

Outros:

1 Bomba 12V

1 Driver Ponte H Dupla L298N

1 Broca

1 Tubo PVC (1M)

1 Mangueira (1,5 M)

1 Balde de Plástico (5L)

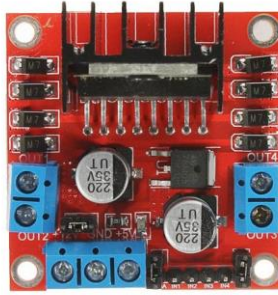
Imagens Equipamentos



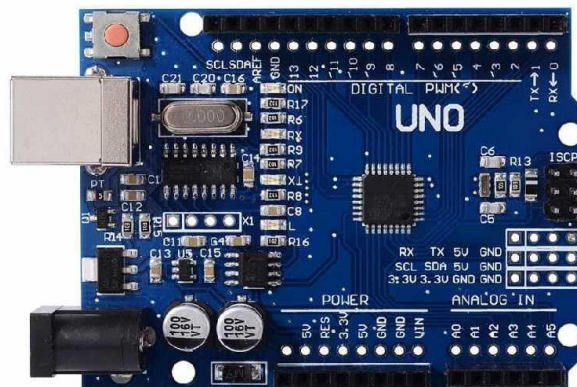
Motor DC



Termômetro Digital DS18B20 e Display LCD



Driver Ponte H Dupla L298N



Arduíno UNO



Balde de Plástico (5L)



Mangueira (1,5 M)



Garrafa Pet (500ml)



Broca

Código do projeto:

```
#include <Blynk.h>
#include <BlynkSimpleStream.h>

#include <FastIO.h>
#include <I2CIO.h>
#include <LCD.h>
#include <LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>
#include <LiquidCrystal_SR.h>
#include <LiquidCrystal_SR2W.h>
#include <LiquidCrystal_SR3W.h>
#include <BlynkSimpleStream.h>
#include <Wire.h>
#include <OneWire.h>
```



```
#include <DallasTemperature.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#define ONE_WIRE_BUS 3
#define BLYNK_PRINT SwSerial
```

```
SoftwareSerial SwSerial(10, 11); // RX, TX
```

```
// Setup a oneWire instance to communicate with any OneWire devices (not just
Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Set the LCD I2C address
```

```
//Codigo Celular Gustavo: e9eef5c379d04bf5801021b17e04dfc3
//Codigo Celular Diego: b96f44be9b8e4c21b4e615c38f1c6973
char auth[] = "b96f44be9b8e4c21b4e615c38f1c6973";
```

```
int IN1 = 11;
int IN2 = 6;
int IN3 = 5;
int IN4 = 10;
```

```
void setup()
{
```

```
    //Define os pinos como saida
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
```

```
    // start serial port
    // Serial.begin(9600);
    // Serial.println("Dallas Temperature IC Control Library Demo");
    // Start up the library
    sensors.begin();
    lcd.begin(16,2);
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0,0);
```

```
    // Debug console
    SwSerial.begin(9600);
```



```

// Blynk will work through Serial
// Do not read or write this serial manually in your sketch
Serial.begin(9600);
Blynk.begin(Serial, auth);

}

void loop() {

  Blynk.run();
}

// Attach virtual serial terminal to Virtual Pin V1
WidgetTerminal terminal(V3);

// You can send commands from Terminal to your hardware. Just use
// the same Virtual Pin as your Terminal Widget
BLYNK_WRITE(V3)
{

  // if you type "Marco" into Terminal Widget - it will respond: "Polo:"
  if (String("Marco") == param.asStr()) {
    terminal.println("You said: 'Marco'");
    terminal.println("I said: 'Polo'");
  } else {

    // Send it back
    terminal.print("You said:");
    terminal.write(param.getBuffer(), param.getLength());
    terminal.println();
  }

  // Ensure everything is sent
  terminal.flush();
}

// Botão - Derrama Malte
BLYNK_WRITE(V10){
  //Gira o Motor A
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  delay(2000);
  //Para o motor A
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  delay(500);
}

```

```
    //Botão - Mexer Mosto
BLYNK_WRITE(V5){
    //Gira o Motor B
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(2000);
    //Para o motor B
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
    delay(500);
}
```

```
    //Botão - Mostra Temperatura
BLYNK_WRITE(V2){
    sensors.requestTemperatures(); // Send the command to get temperatures
    // Serial.print("Temperature for the device 1 (index 0) is: ");
    float x = sensors.getTempCByIndex(0);
    // Serial.println(x);
    lcd.setCursor(0,0);
    lcd.print("Temperature: ");
    lcd.setCursor(0,1);
    lcd.print("      ");
    lcd.setCursor(0,1);
    lcd.print(x);

    delay(3000);
}
```