

ESCOLA SENAI “A. JACOB LAFER”

CURSO TÉCNICO EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

LEONARDO BANIN

LAURA OLIVEIRA

FELIPE SILVA

GIOVANNA MAIOLI

FLEX-FOOD

SANTO ANDRÉ

2024

LEONARDO BANIN

LAURA OLIVEIRA

FELIPE SILVA

GIOVANNA MAIOLI

FLEX-FOOD:

Cantina Online

Trabalho de Conclusão de
Curso apresentado à disciplina de
projetos na Escola SENAI “A. Jacob Lafer”,
como requisito à obtenção do grau Técnico
em

Análise e desenvolvimento de
sistemas.

Orientador: Nadja Luz

Co orientador: Raul Lopes

SANTO ANDRÉ

2024

LEONARDO BANIN

LAURA OLIVEIRA

FELIPE SILVA

GIOVANNA MAIOLI

FLEX-FOOD:

Cantina Online

Trabalho de Conclusão de
Curso apresentado à disciplina
de projetos na Escola SENAI “A.
Jacob Lafer”, como requisito à
obtenção do grau Técnico em
Análise e desenvolvimento de
sistemas.

Orientador: Nadja Luz

Co-orientador: Raul Lopes

Professor _____ visto _____

Professor _____ visto _____

Professor _____ visto _____

Professor _____ visto _____

Santo André, 12 de Setembro de 2024

RESUMO

Este Trabalho de Conclusão de Curso apresenta o desenvolvimento do **Flex Food**, um sistema de cantina online projetado para otimizar o atendimento em instituições de ensino, como escolas e universidades. O projeto visa resolver o problema das longas filas nas cantinas, que causam perda de tempo e frustração para alunos e funcionários. Para isso, o sistema permite a realização de pedidos e pagamentos antecipados por meio de um site e um aplicativo móvel, de modo que o usuário possa retirar seu pedido de forma rápida e prática, sem a necessidade de enfrentar filas. O desenvolvimento utiliza tecnologias como React.js, React Native e Node.js, seguindo uma metodologia ágil e cumprindo rigorosas políticas de privacidade e segurança de dados. A viabilidade técnica e econômica do projeto foi analisada, demonstrando que a proposta é exequível e competitiva no mercado. Os resultados obtidos indicam que o **Flex Food** pode aumentar significativamente a eficiência do atendimento e a satisfação dos usuários, proporcionando uma solução inovadora e eficiente para a gestão de cantinas em ambientes educacionais.

Palavras-chave: cantina online; pedidos antecipados; aplicativo móvel.

ABSTRACT

This Final Project presents the development of **Flex Food**, an online canteen system designed to optimize service in educational institutions, such as schools and universities. The project aims to solve the problem of long queues in canteens, which cause time loss and frustration for students and staff. To address this, the system allows for pre-orders and payments through a website and a mobile application, enabling users to pick up their orders quickly and conveniently, without the need to stand in line. The development uses technologies such as React.js, React Native, and Node.js, following an agile methodology and adhering to strict privacy and data security policies. The project's technical and economic feasibility was analyzed, demonstrating that the proposal is feasible and competitive in the market. The results indicate that **Flex Food** can significantly increase service efficiency and user satisfaction, providing an innovative and effective solution for canteen management in educational environments.

Keywords: online canteen; pre-orders; mobile application.

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Justificativa	11
1.2 Problema.....	11
1.3 Objetivos.....	11
1.3.1 Objetivo geral	12
1.3.2 Objetivos específicos	12
2 EMBASAMENTO TEÓRICO	13
2.1 Viabilidade Técnica	13
2.2 Viabilidade Econômica.....	13
3 DESENVOLVIMENTO	14
3.1 Solução Inicial	14
3.1.1 Primeiros protótipos.....	15
3.1.2 Diagrama em blocos do projeto	15
3.1.3 Descritivo de funcionamento do projeto	16
3.1.4 Planejamento – Gráfico de Gantt.....	16
4 RESULTADOS	17
4.1 Diagramas Elétricos e/ou Eletrônicos.....	17
4.1.1 Esquema de Ligação do Circuito Emissor	17
4.1.2 Esquema de Ligação do Circuito Receptor	18
4.2 Memorial de cálculo	19
4.3 Layouts das placas	19
4.4 Desenhos mecânicos.....	19
4.5 Fluxograma da programação	19
4.6 Programação	20
4.7 Lista De Materiais.....	21
4.8 Melhorias	21
5 CONSIDERAÇÕES FINAIS.....	21
REFERÊNCIAS	23
APÊNDICE A – GRÁFICO DE GANTT	24
APÊNDICE B – DIAGRAMA ELETRÔNICO (PROTEUS).....	25
APÊNDICE C – DIAGRAMA ELÉTRICO	26
APÊNDICE D – CÓDIGO DA PROGRAMAÇÃO.....	27
ANEXO A – DIAGRAMA ELÉTRICO MKR FOX 1200	38

ANEXO B – DATASHEET MC7805 01	39
--	-----------

1 INTRODUÇÃO

O presente Trabalho de Conclusão de Curso tem como objetivo o desenvolvimento de um sistema de cantina online chamado Flex Food para escola/universidade. O projeto nasceu da necessidade de dar solução a um problema recorrente em ambientes escolares: as grandes filas nas cantinas que geram perda de tempo e insatisfação entre os alunos e os funcionários. O Flex Food propõe a criação de um aplicativo móvel e um site para que os usuários possam solicitar seus lanches e efetuar pagamento com antecedência. Na hora de retirar, o processo será feito por meio de um código gerado pelo sistema, sem a necessidade de enfrentar filas.

O sistema será desenvolvido utilizando tecnologias modernas como Html, Css e JavaScript para o site, AppGyver para o aplicativo móvel, com o intuito de garantir uma interface amigável, responsiva e segura. Este documento abordará desde a concepção do problema, justificativas, levantamento de requisitos e planejamento até a implementação, testes e análise de viabilidade técnica e econômica do projeto. A equipe de desenvolvimento é composta por quatro membros, que utilizarão uma abordagem ágil para garantir o sucesso do projeto dentro do prazo de seis meses estipulado.

1.1 Justificativa

O projeto Flex Food é uma solução rápida para melhorar a compra e venda de alimentos nas cantinas escolares. As pessoas na escola e no trabalho muitas vezes passam muito tempo esperando para conseguir lanches, o que as torna menos produtivas e desanimadas. É sobre o plano da Flex Food para economizar tempo permitindo que as pessoas façam pedidos e paguem on-line antes do intervalo e do almoço.

1.2 Problema

O problema que o plano Flex Food visa servir é o tempo exagerado gasto em filas para a encomenda de lanches nas cantinas de escolas e universidades. Esse ato gera desfastio entre alunos e funcionários, resultando em uma atmosfera de fracasso e desaproveitamento de tempo, notadamente durante horários de pico, quão intervalos e almoços. A falta de uma solução eficiente para organizar e agilizar o seguimento de encomenda de alimentos é uma competição que afeta tão a experimento do utente quão a gestão das cantinas. A necessidade de melhorar a experiência dos consumidores e otimizar o fluxo de trabalho das cantinas motiva o desenvolvimento de uma plataforma que integre pedidos antecipados, pagamentos digitais e retirada rápida.

1.3 Objetivos

Os objetivos do projeto foram determinados com base na identificação da problemática das filas em cantinas e na busca por uma solução eficiente que otimize o atendimento, melhore a experiência dos usuários e facilite a gestão dos estabelecimentos. O projeto visa oferecer um sistema inovador que não apenas

atenda à demanda atual, mas que também seja escalável e adaptável a diferentes contextos de instituições de ensino.

1.3.1 Objetivo geral

Desenvolver um sistema de cantina online, denominado **Flex Food**, que possibilite aos usuários realizarem pedidos e pagamentos antecipados de lanches em escolas e universidades, reduzindo o tempo de espera em filas e melhorando a eficiência do atendimento e a satisfação dos clientes.

1.3.2 Objetivos específicos

- Implementar uma interface web utilizando React.js que permita a visualização do cardápio, realização de pedidos e pagamentos online de forma intuitiva e segura.
- Desenvolver um aplicativo móvel no AppGyver que ofereça as mesmas funcionalidades da interface web, adaptado para smartphones, garantindo responsividade e usabilidade.
- Garantir a segurança e privacidade dos dados dos usuários, conforme as normas e regulamentos vigentes, utilizando boas práticas de criptografia e segurança de informações.
- Realizar testes de usabilidade, desempenho e segurança para assegurar a qualidade do sistema e promover melhorias contínuas com base no feedback dos usuários.

2 EMBASAMENTO TEÓRICO

O embasamento teórico deste trabalho visa fundamentar o desenvolvimento do projeto **Flex Food** através da análise do cenário, das tecnologias empregadas e da viabilidade técnica e econômica do sistema. A seguir, são apresentados os aspectos teóricos que sustentam a proposta do sistema de cantina online, destacando sua aplicabilidade e justificção.

2.1 Viabilidade Técnica

A viabilidade técnica do **Flex Food** é sustentada pela escolha das seguintes tecnologias:

- **AppGyver**: Plataforma de desenvolvimento low-code que permite a criação de aplicativos móveis e web sem a necessidade de escrever grandes quantidades de código. O uso do AppGyver possibilita o desenvolvimento rápido e eficiente de aplicativos, além de permitir a integração com APIs e serviços externos. A escolha do AppGyver se justifica pela sua capacidade de reduzir o tempo de desenvolvimento e permitir a prototipagem ágil de funcionalidades.
- **HTML**: Linguagem de marcação utilizada para estruturar o conteúdo das páginas web. O HTML será utilizado para criar a estrutura básica do site do **Flex Food**, garantindo a organização adequada dos elementos e a apresentação correta das informações aos usuários.
- **CSS**: Linguagem de estilo utilizada para definir a aparência visual das páginas web. O CSS será utilizado para estilizar o site, garantindo que a interface seja atraente e funcional. A escolha do CSS permite a personalização completa do layout e a criação de uma experiência de usuário coesa e responsiva.
- **JavaScript**: Linguagem de programação que permite a criação de funcionalidades dinâmicas nas páginas web. O JavaScript será utilizado para implementar interações e lógica de negócios no site, como o processamento de formulários de pedido. A escolha do JavaScript é fundamentada em sua capacidade de criar aplicações interativas e melhorar a experiência do usuário.

2.2 Viabilidade Econômica

A viabilidade econômica do projeto **Flex Food** é analisada com base nos recursos necessários para sua implementação e na comparação com soluções existentes no mercado. Considerando que o projeto não incluirá um backend, os principais aspectos são:

- **Recursos Materiais e Humanos**: O desenvolvimento do **Flex Food** requer a contratação de desenvolvedores com conhecimento em HTML, CSS e

JavaScript, além de um especialista em AppGyver para garantir a utilização eficiente da plataforma low-code. Será necessário também investir em ferramentas de desenvolvimento e teste, além de um plano de manutenção para o site e o aplicativo.

- **Custos:** O investimento inicial inclui despesas com a licença do AppGyver (se aplicável), ferramentas de desenvolvimento e serviços de hospedagem. Considerando que não haverá backend, os custos relacionados a servidores e serviços de backend são eliminados. A manutenção contínua incluirá atualizações de software e suporte técnico.
- **Análise de Mercado:** O **Flex Food** atende à demanda específica de instituições educacionais que buscam otimizar o processo de compra de lanches. O mercado para soluções de automação em ambientes educacionais está em expansão, e o **Flex Food** se destaca por oferecer uma solução acessível e integrada, em comparação com sistemas existentes que podem ser mais complexos e caros.
- **Competitividade e Projeção de Vendas:** A proposta do **Flex Food** é competitiva devido à sua abordagem prática e ao uso de tecnologias de baixo custo e fácil implementação. A projeção de vendas deve considerar a adesão de instituições de ensino e a possibilidade de expansão para outras áreas. A análise de custo-benefício deve levar em conta a economia de tempo e a melhoria na experiência dos usuários, que são fatores-chave para a aceitação do sistema.

A viabilidade econômica do **Flex Food** é positiva, considerando o potencial de retorno sobre o investimento e a necessidade crescente de soluções eficientes no setor educacional. O projeto oferece uma alternativa acessível e eficaz para melhorar a gestão das cantinas e a satisfação dos usuários, sem a complexidade e os custos associados a um backend.

3 DESENVOLVIMENTO

Elaborar um texto contendo a metodologia para o desenvolvimento do seu projeto, quais etapas foram seguidas para preposição da ideia deste projeto.

3.1 Solução Inicial

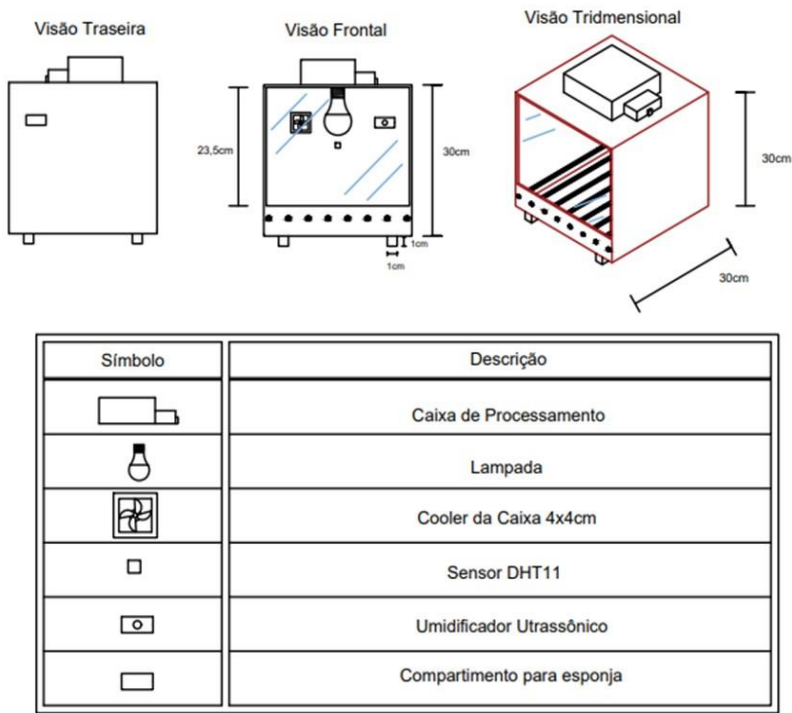
Elabore um texto relatando as etapas para chegar na solução e qual a estratégia adotada pelo grupo para alcançar os objetivos propostos.

3.1.1 Primeiros protótipos

Elaborar um texto descrevendo e explicando o que é o projeto, e se possível apresente na forma de um croqui, esboço ou foto do protótipo do projeto a ser construído conforme o exemplo da figura 1.

Pode ser foto ilustrativa do painel ou caixa contendo o projeto.

Figura 1 – Croqui do projeto (exemplo).



Fonte: Elaborado pelos autores (2024).

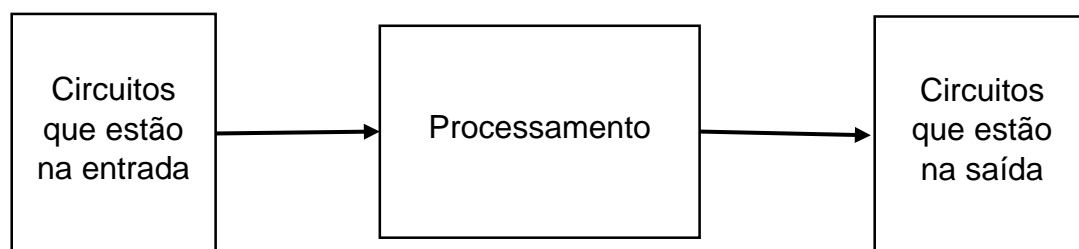
3.1.2 Diagrama em blocos do projeto

Elaborar um texto que explique o diagrama em blocos do projeto a ser construído, identificando quais são elementos a serem considerados.

Inserir a figura do diagrama em blocos. Conforme a figura 2

Sugestão: Entradas a esquerda, processo no meio e saídas à direita, todos identificados e pode conter figuras.

Figura 2 – Diagrama em blocos do projeto(exemplo).



Fonte: Elaborado pelos autores (2024).

3.1.3 Descritivo de funcionamento do projeto

Descrever aqui, de forma detalhada, como funciona o projeto. Detalhar ao máximo toda operação, imagine que você está sentado na frente do protótipo e conforme você for operando o mesmo, você descreve como ele funciona.

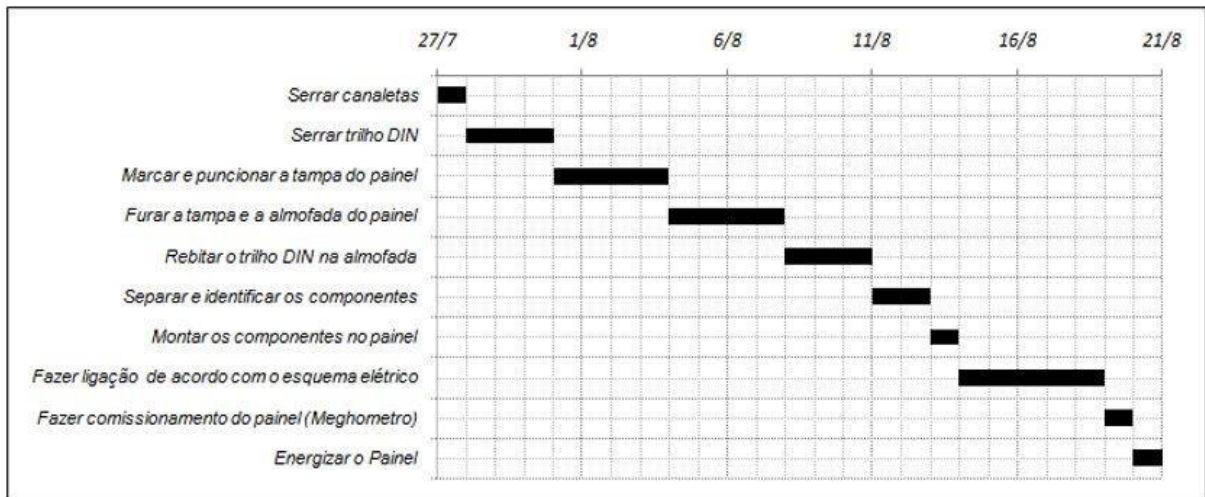
(DESCRITIVO DE OPERAÇÃO)

3.1.4 Planejamento – Gráfico de Gantt

Aqui deve-se elencar em um texto contendo as atividades e responsáveis para a execução do projeto. Elaborar o gráfico de Gantt e insira como um quadro para ilustrar as etapas e prazos sugeridos pelo grupo. Conforme o exemplo do quadro1.

O gráfico completo deve ser colocado como Apêndice, após as referências.

Quadro 1 – Planejamento – Gráfico de Gantt (exemplo).



Fonte: Elaborado pelos autores (2024).

ENTREGA DA DOCUMENTAÇÃO PARCIAL

Ao término dos itens acima deve ser feita uma entrega em formato eletrônico (doc e pdf) para pontuação do grupo na data indicada pelo orientador do projeto.

4 RESULTADOS

Elabore um texto que relate o que será demonstrado neste capítulo.

4.1 Diagramas Elétricos e/ou Eletrônicos

Elabore um texto que relate o que será demonstrado aqui, devendo ter imagens dos principais diagramas elétricos/eletrônicos elaborados. Os diagramas completos podem ser inseridos como demonstrado no APÊNDICE B e APÊNDICE C.

4.1.1 Esquema de Ligação do Circuito Emissor

Elabore um texto que explique o que será demonstrado aqui.

4.2 Memorial de cálculo

Elabore um texto que relate o que será demonstrado aqui bem como a demonstração de todos os cálculos executados no projeto.

4.3 Layouts das placas

Elabore um texto que relate o que será demonstrado aqui bem como a imagem das placas elaboradas.

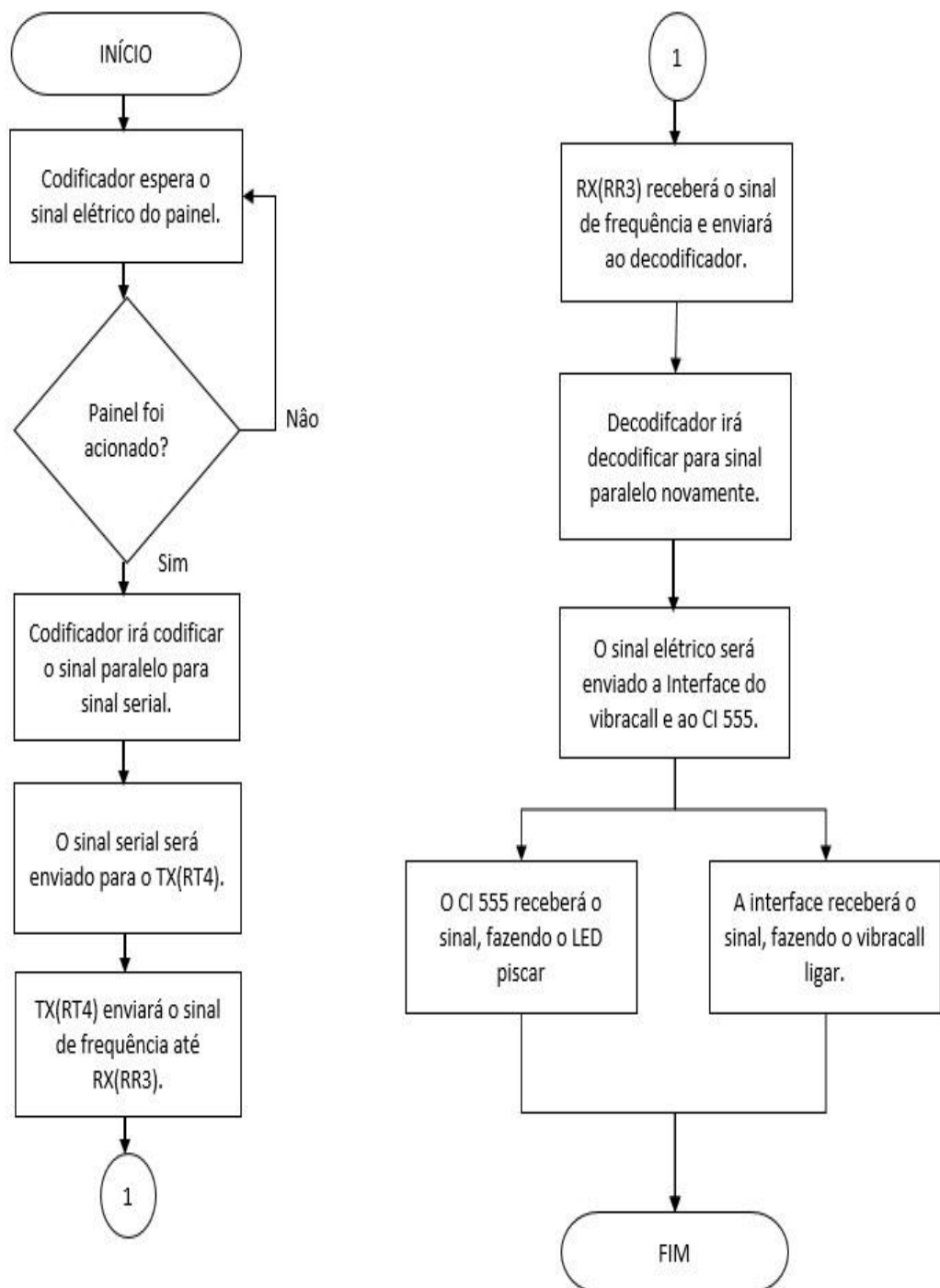
4.4 Desenhos mecânicos

Elabore um texto que relate o que será demonstrado aqui bem como todos os desenhos mecânicos elaborados no projeto.

4.5 Fluxograma da programação

Elabore um texto que relate o que será demonstrado aqui e insira a imagem do fluxograma da programação do projeto.

Figura 4 – Fluxograma (exemplo).



Fonte – Elaborado pelos autores (2024).

4.6 Programação

Elabore um texto que explique detalhes da sua programação: declaração de variáveis, rotinas ou códigos específicos que fazem o seu projeto funcionar. Pode demonstrar aqui dando destaque com imagens ou figuras de recortes do programa que exemplificam. O programa na íntegra deve ser colocado como apêndice após as referências, conforme o APÊNDICE D(exemplo).

Figura 5 – Declaração de variáveis (exemplo).

```
int temperatura = 25;
float umidade = 0.6;
char estadoLuz = 'L';

void setup() {
    // código de inicialização
}

void loop() {
    // código de execução
}
```

Fonte – Elaborado pelos autores (2024).

4.7 Lista De Materiais

Elabore um texto que relate o que será demonstrado aqui bem como a lista completa dos materiais do projeto com os custos.

4.8 Melhorias

Indique quais melhorias ainda podem ser implementadas no projeto.

5 CONSIDERAÇÕES FINAIS

Faça uma breve análise técnica do projeto comparando o planejado com o executado.

Elabore um texto explanando tudo o que você adquiriu de conhecimento com o projeto indicando, também, suas dificuldades e evoluções técnicas.

REFERÊNCIAS

ALLDATASHEET. Electronic Components Datasheet Search: MC7805. **[Site] Alldatashhet.com**. Disponível em: <https://www.alldatasheet.com/datasheetpdf/pdf/82834/FAIRCHILD/MC7805.html>. Acesso em: 27 ago. 2024.

ARAUJO, A. J. N. de. **Análise dos fatores de perdas nos sistemas fotovoltaicos conectados à rede Elétrica em Curitiba**. 2016. Disponível em: http://repositório.roca.utpr.edu.br/jspui/bitstream/1/11997/1/CT_. Acesso em: 27 ago. 2024.

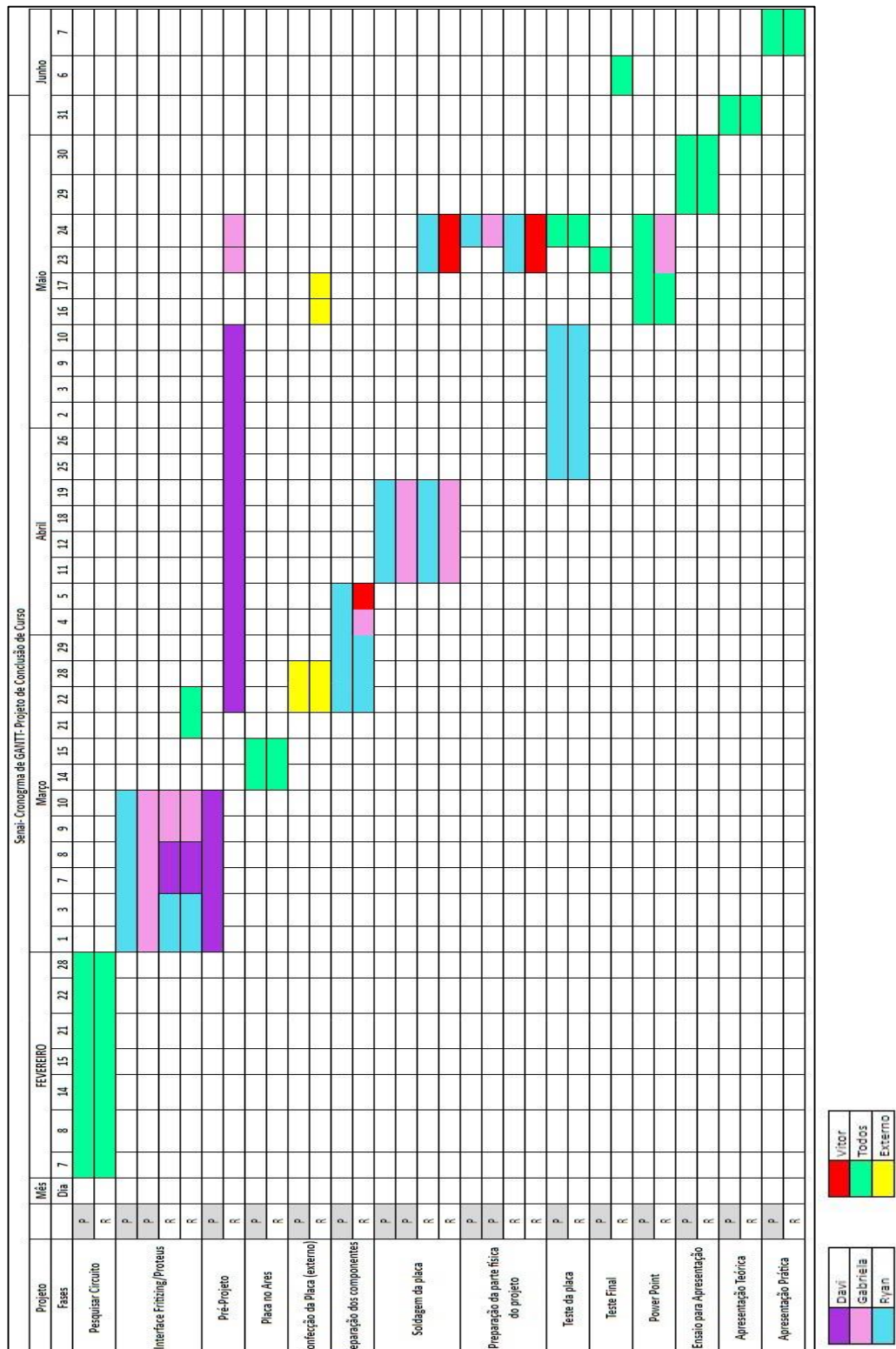
ARDUINO. Schematics for MKR FOX 1200. **[Site] Docs.Arduino.cc**. Disponível em: <https://docs.arduino.cc/resources/schematics/ABX00014-schematics.pdf>. Acesso em: 27 ago. 2024.

CAPUANO, F. G.; IDOETA, I. V. **Elementos de eletrônica digital**. São Paulo: Érica, 2011.

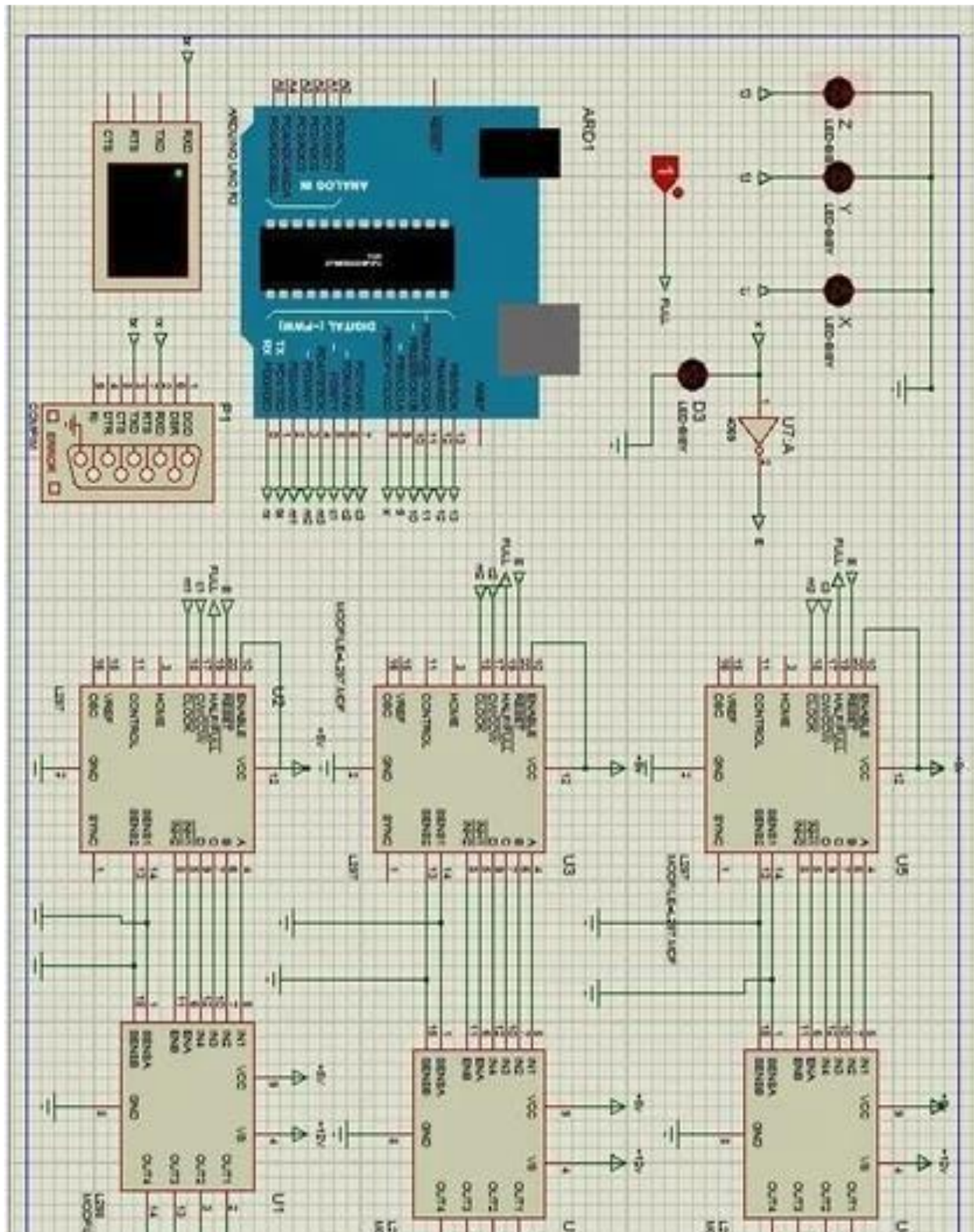
CHIAVENATO, I. **Treinamento e desenvolvimento de recursos humanos**: como incrementar talentos na empresa. 7. ed. Barueri: Manole, 2010. (Série Recursos Humanos).

SOARES, Fernando. Como levar Internet às áreas remotas do Brasil. **[Site] GHZ Ciência e Tecnologia**. [Porto Alegre], 4 set. 2020. Disponível em: <https://gauchazh.clicrbs.com.br/tecnologia/noticia/2020/09/como-levar-internet-asareas-remotas-do-brasil-ckeogzcqb000e014yro2qgssw.html>. Acesso em: 16 mar. 2022. Acesso em: 8 set. 2020.

APÊNDICE A – GRÁFICO DE GANTT



APÊNDICE B – DIAGRAMA ELETRÔNICO (PROTEUS)



APÊNDICE D – CÓDIGO DA PROGRAMAÇÃO

```
#include <twilio.hpp>
#include <WiFi.h>
#define BT1 18
#define BT2 19
#define BT3 21
#define BT4 22
#define BT5 23
#define BT6 25
#define BT7 26
#define BT8 27
#define BT9 32
#define LED 5

bool aux1 = 1; bool aux2 = 1; bool aux3 = 1;
bool aux4 = 1; bool aux5 = 1; bool aux6 = 1;
bool aux7 = 1; bool aux8 = 1; bool aux9 = 1;
bool aux_sistema=0; static const char
*mensagem1 = "JOGAR "; static const char
*mensagem2 = "BRINCAR "; static const
char *mensagem3 = "BEBER AGUA "; static
const char *mensagem4 = "COMER "; static
const char *mensagem5 = "TRISTE"; static
const char *mensagem6 = "DORMIR ";
static const char *mensagem7 =
"BANHEIRO "; static const char
*mensagem8 = "DOR";

static const char *mensagem9 = "IRRITADO ";
```

```
static const char *mensagem_sistema = "SISTEMA INICIADO ";
static const char *Para_Numero = "+5511912715434";//PARA O
NUMERO static const char *Do_Numero = "+12056277820";//DO
NUMERO
```

```
String response;
```

```
static const char *ssid = "iPhone";
static const char *Senha =
"mateus1234";
```

```
static const char *CONTA_SID = "AC1994b899cb24cf89f9072db35c061b62";
static const char *TOKEN_AUTENTICACAO
="ef013055f90f3f0c3019c2a2e887ac5f";
```

```
Twilio
```

```
*twilio; void
setup() {
  pinMode(BT1, INPUT_PULLDOWN);
  pinMode(BT2, INPUT_PULLDOWN);
  pinMode(BT3, INPUT_PULLDOWN);
  pinMode(BT4, INPUT_PULLDOWN);
  pinMode(BT5, INPUT_PULLDOWN);
  pinMode(BT6, INPUT_PULLDOWN);
  pinMode(BT7, INPUT_PULLDOWN);
  pinMode(BT8, INPUT_PULLDOWN);
  pinMode(BT9, INPUT_PULLDOWN);
  pinMode(LED, OUTPUT);

  Serial.begin(9600);
  Serial.print("Conecte-se à rede Wi-fi ;");
  Serial.print(ssid);
  Serial.println("...");
```

```

    WiFi.begin(ssid, Senha);
    twilio = new Twilio(CONTA_SID, TOKEN_AUTENTICACAO);

}

void loop() {
    if (WiFi.status() != WL_CONNECTED)
{
    WiFi.begin(ssid, Senha);
aux_sistema=1;
    while (WiFi.status() != WL_CONNECTED)
    {
        digitalWrite(LED,
HIGH);    delay(100);
        digitalWrite(LED, LOW);
        delay(100);
        Serial.println(".");
    }

    if ((WiFi.status() == WL_CONNECTED) && (aux_sistema==1))
    {
        Serial.println("Conectado!");
        //bool          success          =          twilio->send_message(Para_Numero,
Do_Numero,mensagem_sistema , response);
        digitalWrite(LED,
HIGH);
        aux_sistema=0;
        bool success=1;

        if (success)
        {
            Serial.println("Mensagem Enviada com Sucesso!");

```

```

}
el
se
{
    Serial.println(response);
}
}
}

if ((digitalRead(BT1) == 1) && (aux1 == 1))
{
    Serial.println("Botao
acionado 1");    aux1 = 0;
    delay(100);
    bool success    =    twilio->send_message(Para_Numero,
Do_Numero, mensagem1,response);
    if (success)
    {
        Serial.println("Mensagem Enviada com Sucesso!");

    }
el
se
{
    Serial.println(response);
}
}
if ((digitalRead(BT1) == 0) && (aux1 == 0))
{

```

```

    aux1 = 1;
delay(100);
}
//////////
if ((digitalRead(BT2) == 1) && (aux2 == 1))
{
    Serial.println("Botao
acionado 2");    aux2 = 0;
delay(100);
    bool success    =    twilio->send_message(Para_Numero,
Do_Numero, mensagem2,response);
    if (success)
    {
        Serial.println("Mensagem Enviada com Sucesso!");
    }
    el
    se
    {
        Serial.println(response);
    }
}
if ((digitalRead(BT2) == 0) && (aux2 == 0))
{
    aux2 = 1;
delay(100);
}
//////////
if ((digitalRead(BT3) == 1) && (aux3 == 1))
{

```

```

    Serial.println("Botao
acionado 3");    aux3 = 0;
delay(100);
    bool    success    =    twilio->send_message(Para_Numero,
Do_Numero, mensagem3,response);
    if (success)
    {
        Serial.println("Mensagem Enviada com Sucesso!");
    }
    el
    se
    {
        Serial.println(response);
    }
}
if ((digitalRead(BT3) == 0) && (aux3 == 0))
{
    aux3 = 1;
delay(100);
}
////////////////////
if ((digitalRead(BT4) == 1) && (aux4 == 1))
{
    Serial.println("Botao acionado 4");
aux4 = 0;

    delay(100);

    bool success    =    twilio->send_message(Para_Numero,
Do_Numero, mensagem4,response);

```



```

    if (success)
    {
        Serial.println("Mensagem Enviada com Sucesso!");
    }
    else
    {
        Serial.println(response);
    }
}

if ((digitalRead(BT4) == 0) && (aux4 == 0))
{
    aux4 = 1;
    delay(100);
}

//////////

if ((digitalRead(BT5) == 1) && (aux5 == 1))
{
    Serial.println("Botao
acionado 5");    aux5 = 0;
    delay(100);
    bool success    =    twilio->send_message(Para_Numero,
Do_Numero, mensagem5,response);
    if (success)
    {
        Serial.println("Mensagem Enviada com Sucesso!");
    }
}
else
{
    Serial.println(response);
}
}

```

```

        {
            Serial.println(response);
        }
    }
    if ((digitalRead(BT5) == 0) && (aux5 == 0))
    {
        aux5 = 1;
        delay(100);
    }
    //////////////////////////////////////
    if ((digitalRead(BT6) == 1) && (aux6 == 1))
    {
        Serial.println("Botao
acionado 6");    aux6 = 0;
        delay(100);
        bool    success    =    twilio->send_message(Para_Numero,
Do_Numero, mensagem6,response);
        if (success)
        {
            Serial.println("Mensagem Enviada com Sucesso!");
        }
    }
    else
    {
        Serial.println(response);
    }
}
if ((digitalRead(BT6) == 0) && (aux6 == 0))
{

```

```

    aux6 = 1;
delay(100);
}
//////////
if ((digitalRead(BT7) == 1) && (aux7 == 1))
{
    Serial.println("Botao
acionado 7");    aux7 = 0;
delay(100);
    bool    success    =    twilio->send_message(Para_Numero,
Do_Numero,
mensagem7,response);
    if (success)
    {
        Serial.println("Mensagem Enviada com Sucesso!");
    }
    el
    se
    {
        Serial.println(response);
    }
}
if ((digitalRead(BT7) == 0) && (aux7 == 0))
{
    aux7 = 1;
delay(100);
}
//////////
if ((digitalRead(BT8) == 1) && (aux8 == 1))
{

```

```

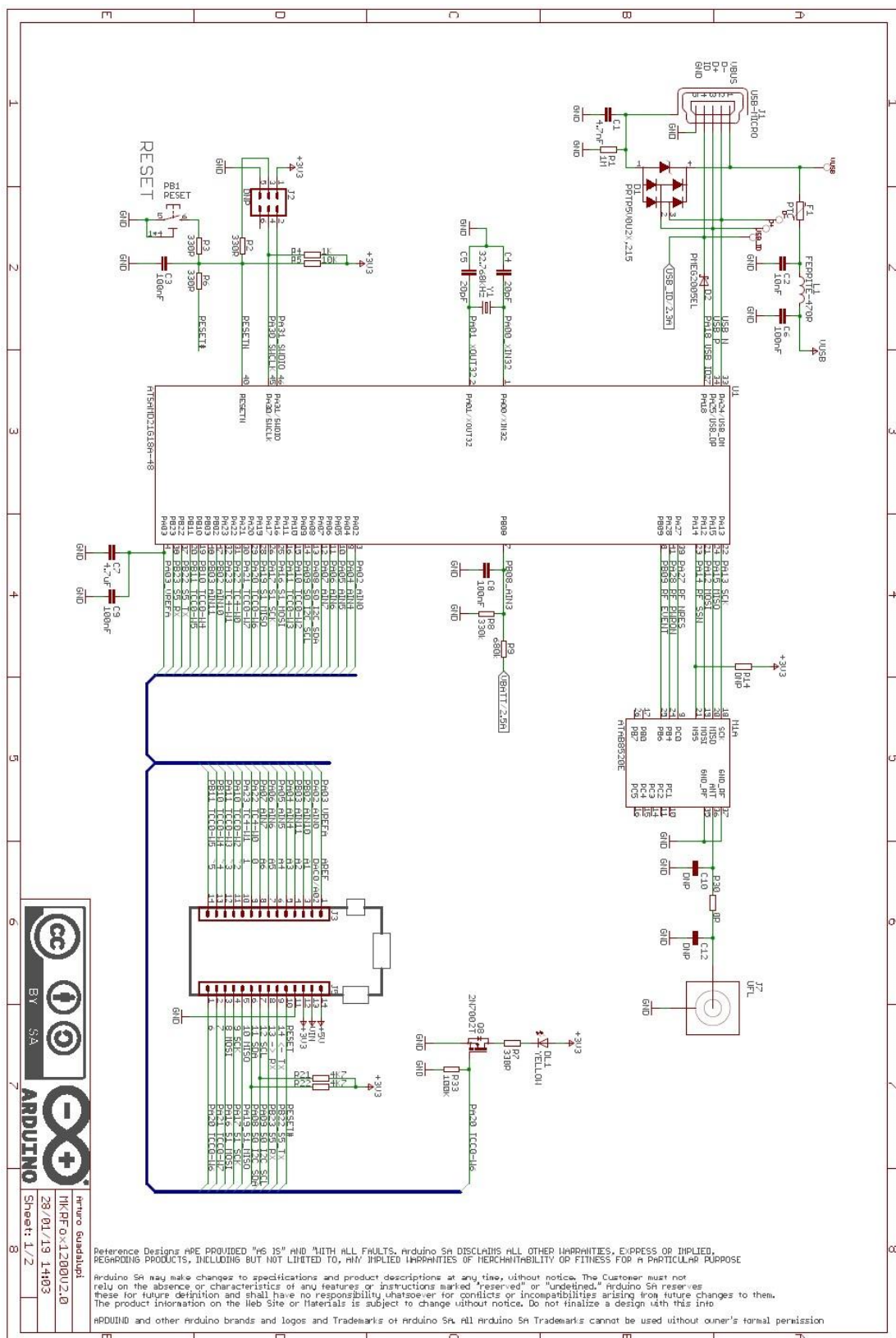
        Serial.println("Botao
acionado 8");    aux8 = 0;
delay(100);
        bool success      =      twilio->send_message(Para_Numero,
Do_Numero, mensagem8,response);
        if (success)
        {
            Serial.println("Mensagem Enviada com Sucesso!");

        }
    el
    se
    {
        Serial.println(response);
    }
}
    if ((digitalRead(BT8) == 0) && (aux8 == 0))
{
    aux8 = 1;
delay(100);
}
//////////
    if ((digitalRead(BT9) == 1) && (aux9 == 1))
    {
        Serial.println("Botao
acionado 9");    aux9 = 0;
delay(100);
        bool      success      =      twilio->send_message(Para_Numero,
Do_Numero, mensagem9,response);
        if (success)
        {

```

```
    Serial.println("Mensagem Enviada com Sucesso!");  
  
}  
el  
se  
    {  
        Serial.println(response);  
    }  
}  
if ((digitalRead(BT9) == 0) && (aux9 == 0))  
{  
    aux9 = 1;  
    delay(100);  
}  
}
```

ANEXO A – DIAGRAMA ELÉTRICO MKR FOX 1200



Fonte: Arduino (2024).

ANEXO B – DATASHEET MC7805 01


www.fairchildsemi.com

MC78XX/LM78XX/MC78XXA

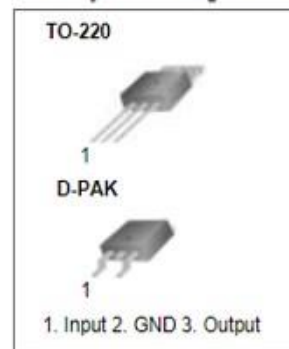
3-Terminal 1A Positive Voltage Regulator

Features

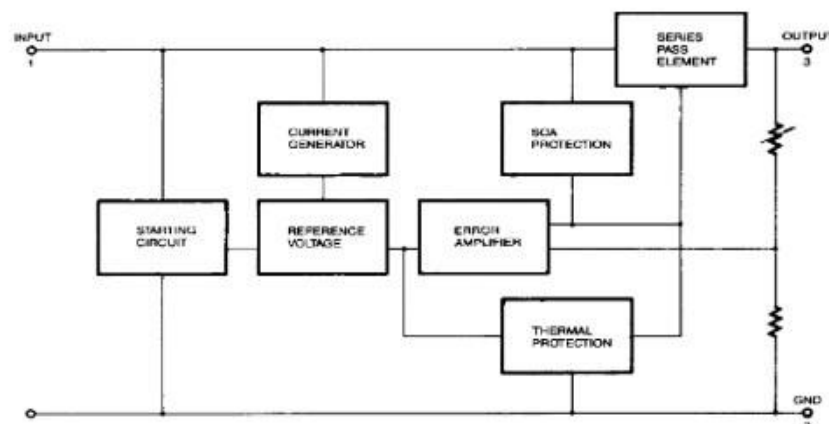
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



Fonte: Alldatasheet (2024).