



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA



PEDRO HENRIQUE DE ALBUQUERQUE GOMES

**Sistema de automação para mercado com múltiplos vendedores
baseado em ESP8266**

Recife

2023

PEDRO HENRIQUE DE ALBUQUERQUE GOMES

**Sistema de automação para mercado com múltiplos vendedores
baseado em ESP8266**

Versão Original

Trabalho de Conclusão de Curso apresentado ao Departamento de Eletrônica e Sistemas como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia Eletrônica.

Orientador: Prof. João Marcelo Xavier Natário Teixeira

Recife

2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Albuquerque Gomes, Pedro Henrique de.

Sistema de automação para mercado com múltiplos vendedores baseado em
ESP8266 / Pedro Henrique de Albuquerque Gomes. - Recife, 2023.

46 p. : il., tab.

Orientador(a): João Marcelo Xavier Natário Teixeira

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de
Pernambuco, Centro de Tecnologia e Geociências, Engenharia Eletrônica -
Bacharelado, 2023.

Inclui referências, apêndices.

1. ESP8266. 2. HTTPS. 3. Arduino. 4. Mercado Autônomo. 5. IoT. I. Xavier
Natário Teixeira, João Marcelo . (Orientação). II. Título.

620 CDD (22.ed.)

Sistema de automação para mercado com múltiplos vendedores baseado em ESP8266

PEDRO HENRIQUE DE ALBUQUERQUE GOMES

Trabalho de Conclusão de Curso apresentado ao Departamento de Eletrônica e Sistemas como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia Eletrônica.

Defendido e aprovado, em 02 de maio de 2023, por:

Prof. João Marcelo Xavier Natário Teixeira
Orientador (DES/UFPE)

Prof. Guilherme Nunes Melo
Coordenador (DES/UFPE)

Recife-PE, 20 de novembro de 2023.

A meus pais, que fizeram por mim mais do que eu poderia agradecer, dedico.

AGRADECIMENTOS

Agradeço inicialmente aos meus pais, Elisabete Albuquerque e Paulo Gomes, e ao meu irmão, Paulo Júnior, por sempre me concederem amor e auxílio na superação de diversos obstáculos da vida.

A todos os professores pelos ensinamentos durante o curso, em especial ao meu orientador, professor João Marcelo, por toda paciência e dedicação por mim e pelo Departamento de Eletrônica e Sistemas.

À Dipolum Consultoria, por ter acreditado no meu potencial e ter me auxiliado durante todo meu desenvolvimento profissional, durante a graduação, em especial aos fundadores Matheus, Angélica, Karl, Bianca e Thaysa.

A todos os amigos que fiz, especialmente do grupo VKNITB que formaram uma grande rede de suporte durante a pandemia, sendo uma peça chave no quesito motivação.

A todos que contribuíram direta ou indiretamente para a realização deste trabalho.

"Ninguém sabe porque chega por último, mas quem vence sabe porque chegou em primeiro."

(Felipe Ret)

RESUMO

GOMES, Pedro. **Sistema de automação para mercado com múltiplos vendedores baseado em ESP8266. 2023. 46f.** Trabalho de Conclusão de Curso (Engenharia Eletrônica) - Universidade Federal de Pernambuco, Recife, 2023.

Com a popularização da internet, os negócios têm se tornado cada vez mais globalizados, sendo, hoje em dia, comum comprar produtos de vendedores de outros estados e até países. Todo esse processo tem acelerado exponencialmente o que chamamos de *marketplace*, um tipo de empresa que conecta oferta e demanda. Esse nicho de mercado foi essencial durante a pandemia, visto que é uma ótima opção para as empresas inserirem seus produtos no mercado digital, com um custo financeiro e intelectual muito baixo. Outro conceito importante que tem se difundido bastante é o de mercado autônomo, um tipo de negócio que utiliza da confiança no cliente para reduzir custos de infraestrutura através de tecnologia, a exemplo de caixas de auto atendimento em super mercados. Esse trabalho traz uma visão disruptiva do mercado, mesclando os conceitos de mercado autônomo e *marketplace* físico, desenvolvendo um sistema de automação para mercado, no qual pequenos vendedores podem cadastrar seus produtos e terceirizar suas vendas presenciais. É proposto um sistema de vendas com produtos e instruções sendo apresentadas em uma tela LCD, gerenciada por um Arduino e conectada à internet através de uma Wemos, na qual executa todo o processo de monitoramento e transferência dos pagamentos para os respectivos vendedores, utilizando um *gateway* de pagamento via HTTPS.

Palavras-chave: ESP8266, HTTPS, Arduino, Mercado Autônomo, IoT.

ABSTRACT

Gomes, Pedro. **Multi-vendor market automation system based on ESP8266. 2023. 46f.** Undergraduate thesis (Electronics Engineering) - Federal University of Pernambuco, Recife, 2023.

With the popularization of the internet, business has become increasingly globalized, and nowadays it is common to buy products from sellers in other states and even countries. This whole process has exponentially accelerated what we call marketplace, a type of company that connects supply and demand. This market niche was essential during the pandemic, as it is a great option for companies to enter their products in the digital market, at a very low financial and intellectual cost. Another important concept that has been widespread is the autonomous market, a type of business that uses customer confidence to reduce infrastructure costs through technology, such as self-service boxes in super markets. This work brings a disruptive view of the market, merging the concepts of an autonomous market and a physical marketplace, developing an automation system for the market, in which small sellers can register their products and outsource their face-to-face sales. A sales system is proposed with products and instructions being displayed on an LCD screen, managed by an Arduino and connected to the internet through a Wemos, in which it performs the entire process of monitoring and transferring payments to the respective sellers, using a gateway of payment via HTTPS.

Keywords: ESP8266, HTTPS, Arduino, Autonomous market, IoT.

LISTA DE ILUSTRAÇÕES

1	Padrão de código de status HTTP [1].	20
2	Protocolo de autenticação e criptografia - TLS [2].	23
3	Esquemático da conexão UART [3].	25
4	Esquema da diferença de tensão durante a transmissão de dados [3]. .	26
5	Ligação entre dispositivo principal e subordinados no barramento SPI [4].	27
6	Princípio de funcionamento de um LCD [5].	28
7	Estrutura do LCD TFT [5].	28
8	Kit Open-Smart One R3 Air utilizado no projeto [6].	31
9	Placa de desenvolvimento Wemos D1 Mini utilizada no projeto [7]. .	32
10	Protótipo de baixa fidelidade do Honest.io	33
11	Protótipo de alta fidelidade do Honest.io	37
12	Honest.io exposto no ArduinoDay	39
13	Tela principal da interface gráfica	45
14	Tela de explicação do sistema	45
15	Tela de QRCode de pagamento	46
16	Tela de agradecimento	46

LISTA DE TABELAS

1	Crescimento, medido em acessos, dos principais <i>sites</i> de <i>marketplace</i> do Brasil, por setor [8].	14
2	Produtos negociados na segunda sessão de testes	37
3	Resultado das negociações em um período semanal	38

LISTA DE ABREVIATURAS E SIGLAS

UFPE	Universidade Federal de Pernambuco
CTG	Centro de Tecnologia e Geociências
DES	Departamento de Eletrônica e Sistemas
DA	Diretório Acadêmico
DADEE	Diretório Acadêmico de Engenharia Elétrica
LCD	Liquid Crystal Display
TFT	Thin Film Transistor
TLS	Transport Layer Security
SSL	Secure Sockets Layer
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
SD	Secure Digital
RAM	Random Access Memory
BMP	Bitmap
API	Application Programming Interface
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
I2C	Inter-Integrated Circuit
GPIO	General Purpose Input/Output
TCP-IP	Transmission Control Protocol/Internet Protocol
WPA	Wi-Fi Protected Access
IoT	Internet of Things
USB	Universal Serial Bus

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo	15
1.1.1	Objetivo geral	15
1.1.2	Objetivos específicos	15
1.2	Motivação do trabalho	16
1.3	Estrutura do trabalho	16
1.4	Trabalhos relacionados	17
1.4.1	Desenvolvimento de um protótipo de baixo custo para segurança residencial [9]	17
1.4.2	Sistema de Alerta para controle de consumo de água por serviços domésticos [10]	18
2	CONCEITOS BÁSICOS	19
2.1	O protocolo HTTP	19
2.1.1	Método GET	20
2.1.2	Método POST	21
2.1.3	O protocolo HTTPS	21
2.1.3.1	Protocolo TLS	21
2.1.4	API RESTful	23
2.2	ARDUINO	24
2.3	Comunicação	24
2.3.1	Serial	24
2.3.2	SPI	26

2.4	LCD TFT Touch Screen	27
3	METODOLOGIA	29
3.1	Open-Smart One R3 Air Easy-Plug	29
3.2	Wemos D1 Mini	31
3.3	Asaas	32
3.4	Protótipo	33
3.4.1	Interface Gráfica	34
3.4.2	<i>Gateway</i> de Pagamento	34
4	RESULTADOS	36
4.1	Primeira sessão de testes	36
4.2	Segunda sessão de testes	36
4.3	Terceira sessão de testes	38
5	CONSIDERAÇÕES FINAIS	40
5.1	Conclusões	40
5.2	Sugestões para trabalhos futuros	40
	Referências	42
	Apêndice A	45
A.1	Imagens apresentadas na tela LCD	45

1 INTRODUÇÃO

A pandemia da Covid-19 teve um impacto significativo no crescimento do comércio eletrônico, segundo a Associação Brasileira de Comércio Eletrônico (ABComm), houve um aumento de 68% em 2020, alcançando R\$ 126,3 bilhões e 11% de participação de mercado. Os *marketplaces* têm sido os principais beneficiados deste crescimento, especialmente para pequenas empresas que ainda não vendiam pela internet, este cenário resultou em uma grande quantidade de novos vendedores em canais *online*. Atualmente, a modalidade ocupa 78% do comércio eletrônico B2C (*Business-to-Consumer*) de acordo com o Ebit|Nielsen [11].

Um *marketplace* é uma plataforma digital que conecta compradores e vendedores de produtos ou serviços, permitindo que eles realizem transações comerciais em um único local [12]. Geralmente, o *marketplace* oferece uma ampla variedade de produtos de diferentes vendedores e, em alguns casos, também pode oferecer serviços .

Os vendedores em um *marketplace* se beneficiam da visibilidade e do espaço digital fornecidos por terceiros, o que lhes permite realizar boas negociações e obter lucro. Em contrapartida, eles pagam comissões e contribuem para manter o *marketplace* em funcionamento [12]. No entanto, essa relação tem mais potencial de lucro e menos riscos do que se o vendedor optasse por investir todos os seus recursos em um *e-commerce* próprio.

Na Tabela 1 pode-se observar o crescimento, durante a pandemia, dos principais *sites* e seus respectivos setores, entre eles pode-se observar a Shein e Shopee, ambos *marketplaces*.

Tabela 1: Crescimento, medido em acessos, dos principais *sites* de *marketplace* do Brasil, por setor [8].

Site*	Setor	Acessos em mar./21	Crescimento início da pandemia (Mar/21 vs. Fev/20)
Shopee	Importados	39.051.120	1852,03%
Shein	Importados	6.738.976	620,23%
Menu Dino	Comidas & Bebidas	4.072.762	552,32%
Alfa Concursos	Educação, Livros & Papelaria	4.156.935	422,55%
Hurb	Turismo	17.564.180	360,13%
Zema	Eletrônicos & Eletrodomésticos	1.202.125	263,60%
DrogaRaia	Farmácia & Saúde	15.282.177	373,76%
Direção Concursos	Educação, Livros & Papelaria	3.826.010	243,99%
Cacau Show	Comidas & Bebidas	4.746.711	894,84%
Drogasil	Farmácia & Saúde	10.496.692	269,74%
Drogaria Minas Brasil	Farmácia & Saúde	2.040.477	227,30%
ObraMax	Casa & Móveis	1.097.711	158,13%
Enjoei	Moda & Acessórios	11.752.302	109,02%
Telha Norte	Casa & Móveis	3.603.429	105,70%
Riachuelo	Moda & Acessórios	12.528.628	149,82%

Assim como os *marketplaces*, o desenvolvimento acelerado das tecnologias e dos meios de pagamentos abrem mercado para o que é chamado de mercado autônomo, que nasce da necessidade de comodidade e praticidade dos usuários que acompanham os novos comportamentos do consumidor, adquiridos ao longo da pandemia com o extenso uso do mercado digital [13]. Normalmente, neles é possível chegar a preços melhores e obter uma margem melhor, comparado a um mercado normal, visto que ao utilizar tecnologias de pagamento voltadas para o usuário, há uma redução de capital humano necessitada para manter o ambiente, consequentemente, reduzindo os custos de operação e dos produtos.

Também é interessante ressaltar o crescimento dos mercados honestos, uma vertente do mercado autônomo que utiliza da confiança para reduzir ainda mais os custos de operação. Este tipo de negócio teve seu crescimento iniciado dentro de condomínios e empresas, começando a surgir no Brasil antes da pandemia, ganhando impulso em 2020 e agora consolidando-se. Empresários do setor acreditam que ainda existe um grande potencial de crescimento e preveem um crescimento na casa dos três dígitos e faturamento milionário para os próximos anos [14].

No Recife, podemos observar uma crescente na implementação de mercados honestos, a exemplo da Minha Quintandinha que já se estabelece em condomínios e empresas [15].

Isso justifica um sistema que implementa um mercado honesto com produtos de diversos vendedores, unindo os conceitos de *marketplace* e mercado honesto, conceitos esses que estão em grande crescimento no país, unindo comodidade para os compradores e levando opções vantajosas de vendas para pequenas empresas, que antes não alcançavam esses ambientes. Além disso, o sistema transfere o trabalho de fornecer produtos nos locais de venda para os vendedores, focando-se apenas em gerenciar os pontos e melhorar a experiência do usuário, tendo como principal fonte de renda uma parcela do lucro dos vendedores.

Este sistema foi nomeado de Honest.io inspirado na sua relação com mercado honesto. Já a terminação “.io” é popularmente utilizada em domínios de corporações de tecnologia, a exemplo da Opensea.io, o maior mercado virtual de NFTs [16], e foi utilizada para expressar o viés tecnológico e disruptivo da solução.

1.1 Objetivo

1.1.1 Objetivo geral

Desenvolver uma aplicação em *hardware* de baixo custo que conecte pequenos vendedores à compradores, baseada em comercialização honesta, a fim de promover os produtos de vendedores autônomos em um ambiente físico, antes inacessível por eles.

1.1.2 Objetivos específicos

- Acessar a internet via Wemos e criar requisições HTTPS de saldo e transferência, para o banco Asaas;
- Desenvolver uma aplicação visual através de um Arduino, utilizando uma tela LCD *Touch Screen*;
- Utilizar micro SD, conectada ao sistema, para guardar imagens da aplicação.

1.2 Motivação do trabalho

Há uma necessidade muito grande de compra de almoços e lanches por parte dos estudantes, diariamente. Na universidade, isso cria uma oportunidade de negócio vantajosa para diversos alunos e moradores locais. Esse mercado acaba movimentando bastante dinheiro e mantendo um comércio local ao redor e dentro da universidade.

Visando isso, com apoio da tecnologia, em alguns locais de acesso controlado (como departamentos e diretórios acadêmicos), podem ser instalados pequenos pontos de vendas distribuídos, que ajudem mutualmente o local que está alocado e os alunos, otimizando tempo e recursos.

Também, uma breve pesquisa de mercado sobre os Mercados Autônomos atuais demonstrou que normalmente são mantidos por uma empresa fixa, que gerencia o estoque e controla o sistema de vendas. Dessa forma, não há possibilidade de uma empresa externa vender seus produtos diretamente nas lojas, funcionando como uma loja comum, porém com mais tecnologia. Isso gera indícios de que há mercado para o que está sendo proposto neste trabalho e não está sendo explorado, que seria o mercado autônomo baseado em cooperativismo, unindo diversos vendedores e um intermediador, como um *marketplace* físico.

1.3 Estrutura do trabalho

É apresentado neste trabalho o estudo e desenvolvimento de um sistema de automação para mercado com múltiplos vendedores, mesclando conceitos de *marketplace* e mercado autônomo, buscando automatizar a venda dos produtos, baixo custo e melhor experiência do usuário.

Este trabalho é organizado em capítulos, da seguinte forma:

No Capítulo de Introdução é apresentado, de uma forma geral, o contexto histórico no qual o problema abordado está imerso, os objetivos e motivação do trabalho, bem como alguns trabalhos relacionados importantes para o desenvolvimento deste.

No Capítulo de Conceitos Básicos são abordados todos os conceitos e tecnologias utilizados no projeto, desde os protocolos de comunicação utilizados na internet (HTTP/HTTPS) e físicos (Serial e SPI), como também os microcontroladores Arduino e Wemos utilizados no desenvolvimento do protótipo proposto.

No Capítulo de Metodologia é abordado de forma cronológica o desenvolvimento do protótipo, citando as especificações técnicas dos componentes utilizados e a motivação para seu uso. Ao final, o protótipo é exposto de forma completa, explicando suas partes e funcionalidades.

No Capítulo de Resultados são detalhadas duas seções de testes às quais o protótipo foi submetido, a primeira sendo mais qualitativa, voltada para teste de funcionalidade, realizada através de simulações de clientes, e a segunda mais quantitativa, voltada para teste de resistência e disponibilidade do sistema, realizada através da implementação do protótipo no Diretório Acadêmico do curso de Elétrica da UFPE.

No Capítulo de Considerações Finais são abordadas as conclusões acerca do projeto desenvolvido, bem como as sugestões de trabalhos futuros, visando a melhoria do protótipo e melhor adequação ao problema.

1.4 Trabalhos relacionados

Neste trabalho foi proposto a criação de um protótipo envolvendo Arduino e ESP8266, portanto teve-se a necessidade de pesquisar trabalhos que utilizaram estes tipos de microcontroladores e os mecanismos que eles utilizaram para passar da ideação até a implementação.

1.4.1 Desenvolvimento de um protótipo de baixo custo para segurança residencial [9]

Este trabalho tem como objetivo apresentar a domótica como uma tecnologia acessível para a população em geral, capaz de proporcionar bem-estar social e conforto residencial, além de reduzir o consumo de recursos como água e energia. Utilizando um microcontrolador Arduino UNO e seus componentes, o autor demonstra como é possível desenvolver projetos de segurança residencial.

Este estudo apresenta um processo de segurança residencial, que ao intruso ser detectado, o alarme será acionado e em outro projeto o morador da residência receberá uma mensagem em seu *smartphone* alertando-o que algo errado está acontecendo em sua residência [9].

1.4.2 Sistema de Alerta para controle de consumo de água por serviços domésticos [10]

Este trabalho de pesquisa teve como objetivo projetar e construir o protótipo de um sistema de controle que permitisse medir e monitorar os níveis de água em tanques domésticos. Para isso, foram utilizadas tecnologias como ESP8266, HTTP e MQTT BROKER.

Além disso, as investigações nele expostas respondem às observações feitas nas casas de Trujillo - Peru, que atualmente possuem sistemas de medição de consumo, que são físicos e cujos procedimentos são realizados manualmente e periodicamente, informando os usuários a cada 25 ou 30 dias.

Com a implementação do sistema de alerta, os usuários recebem informes, em seus celulares, sobre os processos de enchimento e esvaziamento de líquidos nos tanques, gerando assim menos custos de produção [10].

Este projeto é bastante interessante e traz uma gama de informações relevantes sobre o processo de comunicação HTTP, utilizando a ESP8266.

2 CONCEITOS BÁSICOS

2.1 O protocolo HTTP

O HTTP, ou *Hypertext Transfer Protocol*, é um protocolo de transferência que possibilita o acesso à informações através de URLs baseadas em texto. Esse sistema é a base da comunicação de toda internet, oficialmente lançado em janeiro de 1997, como HTTP/1.1, definido no RFC 2068 [17].

Ele funciona com base em conexão, ou seja, é necessário estabelecer uma conexão entre o cliente e o servidor, antes de enviar uma requisição, e a conexão pode ser encerrada ou mantida aberta, a depender da configuração dos sistemas envolvidos, sendo possível fazer várias requisições utilizando a mesma conexão [18].

Um servidor, comumente usado para hospedagem de *site*, tem seu sistema projetado para receber solicitações via HTTP e, por sua vez, o navegador se comporta como um cliente HTTP que envia requisições constantemente.

Os processos de comunicação são comumente chamados de métodos e o seu uso pode ser chamado de requisição, os principais métodos são GET e POST, descritos posteriormente.

A mensagem enviada normalmente contém 3 partes: método utilizado, versão do protocolo e o *header*, já a mensagem recebida contém o *status* da mensagem, versão do protocolo, *header* e *body*. A informação de resposta, que é recebida do servidor, contém em seu *header*, informações como data, tamanho, tipo de arquivo e informações sobre o próprio servidor, já no *body* está contido o conteúdo requisitado.

O *status* da resposta é padronizado e pode ser visualizado na Figura 1, os *status* mais comuns são “200 ok” quando a requisição ocorreu com sucesso ou “400 bad request” quando há algum erro na requisição.

HTTP STATUS CODES	
2xx Success	
200	Success / OK
3xx Redirection	
301	Permanent Redirect
302	Temporary Redirect
304	Not Modified
4xx Client Error	
401	Unauthorized Error
403	Forbidden
404	Not Found
405	Method Not Allowed
5xx Server Error	
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

Figura 1: Padrão de código de status HTTP [1].

2.1.1 Método GET

O método GET não contém *body* e é utilizado para solicitar informações ao servidor, caso a informação seja sensível, é normalmente adicionado um *token* de autenticação no *header* [19].

Tem seus conteúdos passados no *header* da requisição e podem ser visualizados no conteúdo da URL, esse fato limita o máximo de dados enviados, no geral, não podendo passar de 255 caracteres. Desta forma, não é comum o uso para realizar mudanças no servidor e sim para consultas.

No projeto em questão, este método é utilizado para validação de pagamentos, realizando requisições recorrentes de saldo, enviando um token de autenticação no *header*.

2.1.2 Método POST

Este método é muito utilizado para envio de informações para o servidor, diferenciando do método anterior por utilizar *body* em suas requisições, desse modo, o conteúdo é enviado no corpo da mensagem, não ficando visível na URL, não necessariamente é mais seguro, mas abre a possibilidade de envio de informações um pouco maiores, como imagens [19].

O fato de poder enviar mais informações, torna o método comumente utilizado para realizar modificações no servidor ou realizar pedidos de execução de processos, passando parâmetros para tal.

No projeto em questão, este método é utilizado para criar transferências via PIX, realizando o pagamento dos vendedores, enviando dados de valor e chave no *body*.

2.1.3 O protocolo HTTPS

O HTTPS surge da necessidade de proteção dos dados transferidos durante as requisições, o protocolo implementa uma camada extra de segurança, denominada criptografia. Os protocolos de transferência de dados criptografados mais utilizados no HTTPS são SSL e TLS, sendo o primeiro legado por apresentar falhas de vulnerabilidade que poderiam expor informações criptografadas, a CISA (Cybersecurity & Infrastructure Security Agency) desenvolveu um alerta sobre a vulnerabilidade do protocolo [20].

2.1.3.1 Protocolo TLS

Denominado de *Transport Layer Security*, teve seu desenvolvimento acelerado nas descobertas de vulnerabilidades do protocolo SSL, proposto pela IETF (Internet Engineering Task Force), uma organização internacional de normas, com sua primeira versão publicada em 1999, está atualmente na versão 1.3, publicada em 2018 [21].

O TLS 1.0 na verdade é o SSL 3.1, porém houve uma mudança de nomenclatura a fim de indicar a desassociação do protocolo a Netscape (desenvolvedora do SSL), é comumente utilizado os termos SSL e TLS quando se fala da camada de criptografia do HTTPS, havendo uma confusão nas nomenclaturas até hoje [22].

Esse protocolo une as forças das chaves assimétricas e simétricas, durante o pro-

cesso de comunicação, partindo de três funções principais: criptografia, autenticação e integridade.

Para utilização do TLS o servidor necessita ter um certificado TLS instalado (também conhecido por certificado SSL), o certificado é emitido por uma autoridade de certificação para a pessoa ou empresa dona do domínio. Ele contém dados sobre o proprietário do domínio e a chave pública do servidor utilizada no processo de encriptação [23].

A conexão TLS é iniciada por uma sequência chamada de *handshake TLS*, observada na Figura 2. Essa sequência realiza entre o cliente e servidor, uma série de trocas de mensagens como alinhamento da versão do TLS que será utilizada, as suítes de cifras, autenticação por meio do certificado e elaboração das chaves utilizadas no processo de criptografia para aquela sessão [24].

Por fim, as requisições e dados são transferidos utilizando o processo de criptografia de chave pública e privada, no qual é possível criptografar os dados com a chave pública e apenas descriptografar com a chave privada, e são assinados com um autenticador de mensagem (MAC), que será usado pelo destinatário para verificação de integridade.

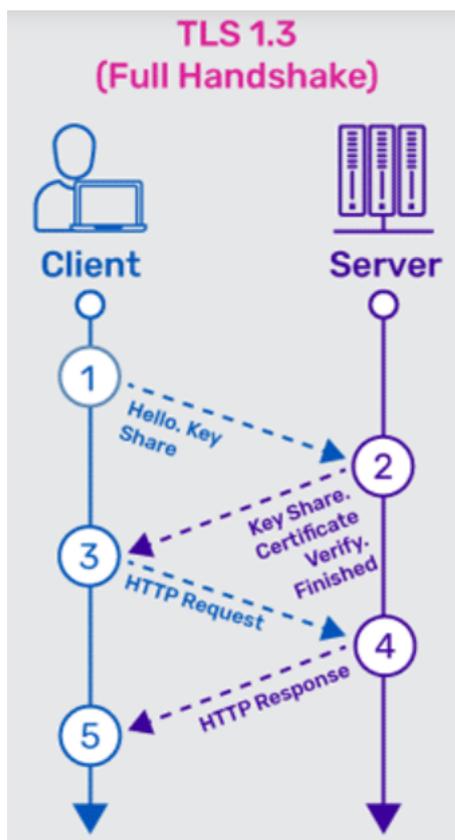


Figura 2: Protocolo de autenticação e criptografia - TLS [2].

2.1.4 API RESTful

É denominado API (*Application Programming Interface*) um conjunto de instruções, protocolos e ferramentas que permite que diferentes aplicações e sistemas possam interagir entre si.

Uma API RESTful segue o estilo arquitetural REST (*Representational State Transfer*) que é um conjunto de princípios e restrições que definem como as aplicações devem ser projetadas e desenvolvidas [25].

As APIs RESTful são baseadas em recursos identificados por uma URI (Uniform Resource Identifier), que é um identificador único na internet, e seus recursos podem ser manipulados através de operações HTTPS, como GET, POST, PUT e DELETE, que correspondem a ações básicas de leitura, criação, atualização e exclusão de recursos [26].

Também são caracterizadas pela separação clara entre cliente e servidor, significando que o servidor não mantém nenhum estado da sessão do cliente, sendo cada

requisição independente.

Outra característica importante é o uso de formatos de dados padronizados para representação dos recursos, sendo o formato mais comum o JSON (*JavaScript Object Notation*) [25].

São projetadas para ser consumidas como um produto, acompanhada de sua documentação e normalmente mantida pelos mantenedores da informação.

2.2 ARDUINO

É uma plataforma de prototipagem eletrônica amplamente utilizada por ser simples e tornar o desenvolvimento de protótipos mais fácil e barato.

Criada em meados de 2005 utilizando conceitos de código aberto, pode ser programada a partir de uma linguagem baseada em C/C++, sendo possível a criação de projetos de robôs, IoT, alarmes e diversos outros [27].

Além da linguagem de programação, a plataforma também conta com uma série de placas de desenvolvimento, que têm como base microcontroladores, equipados com ferramentas de comunicação (Serial, SPI e/ou I2C) e portas I/O digitais e analógicas, tudo isso pode ser programado através da Arduino IDE e carregado via USB [28].

A comunidade formada ao entorno desenvolve e disponibiliza muitos projetos e bibliotecas interessantes, a exemplo da Prusa i3 (projeto de impressora 3D utilizando Arduino Mega).

A maioria das extensões de *hardware* (*shields*) que podem ser comprados separadamente e conectados nas placas de desenvolvimento foram desenvolvidas pela comunidade e têm suas bibliotecas, esquemáticos e exemplos de uso disponíveis na internet.

2.3 Comunicação

2.3.1 Serial

Comunicação serial é um método de transferência de dados em que as informações são enviadas *bit a bit*, um após o outro, através de um canal de comunicação. Existem vários protocolos de comunicação serial, mas um dos mais comuns é o UART

(*Universal Asynchronous Receiver/Transmitter*), que é suportado por muitos microcontroladores, incluindo o Arduino [29].

UART é universal pois permite a conexão entre diversos dispositivos de forma assíncrona, pois não apresenta *clock* para coordenar as ações entre dispositivos, porém precisa-se especificar a taxa de transmissão, normalmente 115200 ou 9600 *baud rates* [30].

O Arduino tem várias portas que podem ser usadas para esse tipo de comunicação, incluindo as portas digitais 0-RX e 1-TX (que são usadas pelo módulo UART integrado) e as portas digitais 2 e 3 (que são usadas pelo módulo de comunicação serial por software, chamado de "SoftwareSerial") [31].

Na Figura 3 é possível observar a conexão UART entre dois dispositivos e a orientação da passagem de dados, utilizando as portas RX e TX.

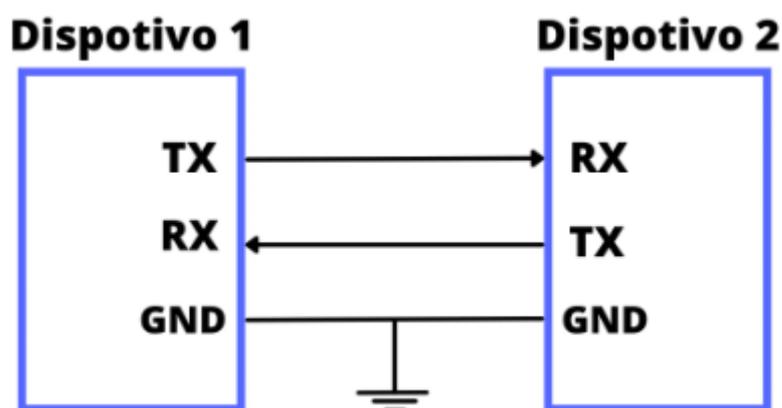


Figura 3: Esquemático da conexão UART [3].

Essas portas também são utilizadas para carregamento de código nos microcontroladores, necessitando remover as conexões sempre que for preciso atualizar os códigos.

O início da transmissão de dados é marcada pela descida do sinal para 0V, logo após, os *bits* são enviados conforme a tabela ASCII. No exemplo a seguir pode ser visto o envio da letra Z (0101 1010, segundo a tabela ASCII) [3].

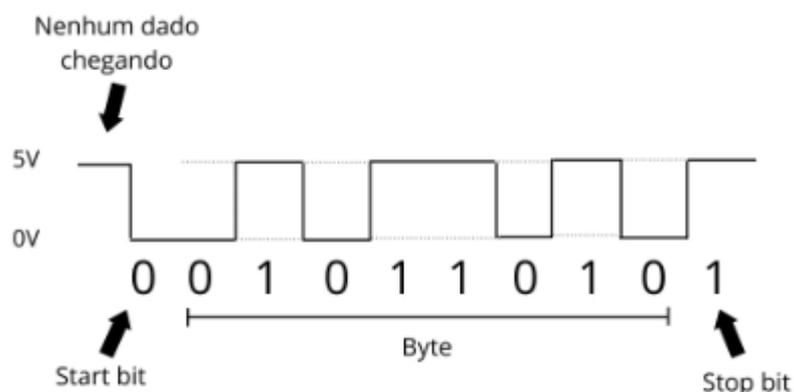


Figura 4: Esquema da diferença de tensão durante a transmissão de dados [3].

No projeto em questão foram utilizadas as portas 0 e 1 para comunicação entre o Arduino e a Wemos, enviando e recebendo dados sobre o *status* de pagamento, valor a ser recebido e saldo da conta.

2.3.2 SPI

A comunicação SPI (*Serial Peripheral Interface*) é uma das formas mais comuns de se comunicar com dispositivos periféricos, usando um microcontrolador. O protocolo permite a comunicação serial síncrona entre dispositivos, onde um dispositivo principal (no caso o Arduino), controla a comunicação com um ou mais dispositivos subordinados [32]. A comunicação ocorre através de quatro sinais principais:

- MOSI (*Master Out Slave In*) - Este pino é usado para enviar dados do dispositivo principal para os subordinados [33];
- MISO (*Master In Slave Out*) - Este pino é usado para enviar dados do dispositivo subordinado para o principal [33];
- SCK (*Serial Clock*) - Este pino é usado para sincronizar a transferência de dados entre o dispositivo principal e subordinado [33];
- SS (*Slave Select*) - Este pino é usado pelo dispositivo principal para selecionar com qual subordinado deseja se comunicar [33].

Para usar a comunicação SPI, com um Arduino, é necessário conectar os pinos correspondentes do dispositivo principal aos pinos correspondentes dos dispositivos

subordinados (por exemplo, sensores ou *displays*), na Figura 5 é possível observar essa conexão.

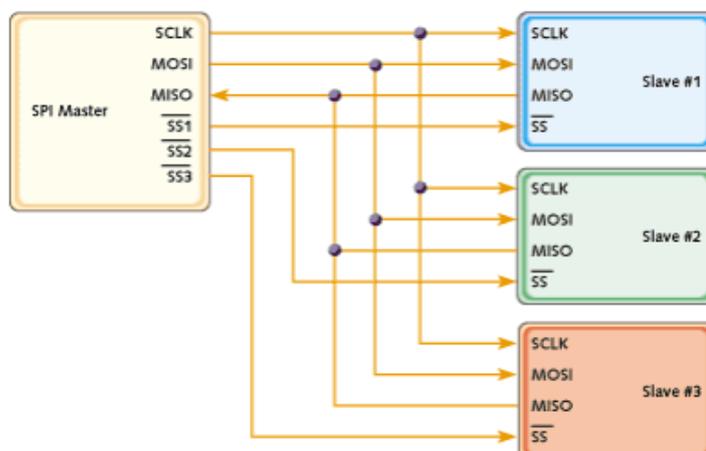


Figura 5: Ligação entre dispositivo principal e subordinados no barramento SPI [4].

No Arduino, a biblioteca SPI permite a configuração dos pinos que serão usados para cada uma das quatro funções principais da comunicação SPI, bem como a definição da taxa de transferência de dados. A função “SPI.transfer()” é usada para enviar e receber dados durante a comunicação SPI.

A comunicação SPI pode ser uma opção útil para se comunicar com dispositivos periféricos que exigem altas taxas de transferência de dados ou que precisam de uma comunicação mais robusta do que a comunicação serial padrão do Arduino.

No projeto em questão, foi utilizada a comunicação SPI para ler as telas do visor (imagens BMP) no cartão SD.

2.4 LCD TFT Touch Screen

Dispositivos LCDs (*Liquid Crystal Display*) são compostos por cristal líquido entre duas folhas de vidro. Seu funcionamento é dado pela orientação das moléculas do cristal líquido que ficam alinhadas paralelamente à superfície do vidro quando não há uma tensão aplicada, mas mudam de direção ficando verticais quando há uma tensão aplicada [34]. Na Figura 6 é possível observar que há uma camada de cristal líquido entre as duas folhas de vidro, com placas polarizadas fixadas. As cores, intensidade e consequentemente as imagens, são formadas pelo jogo de polarização e orientação do cristal líquido, a qual a luz é submetida [34].

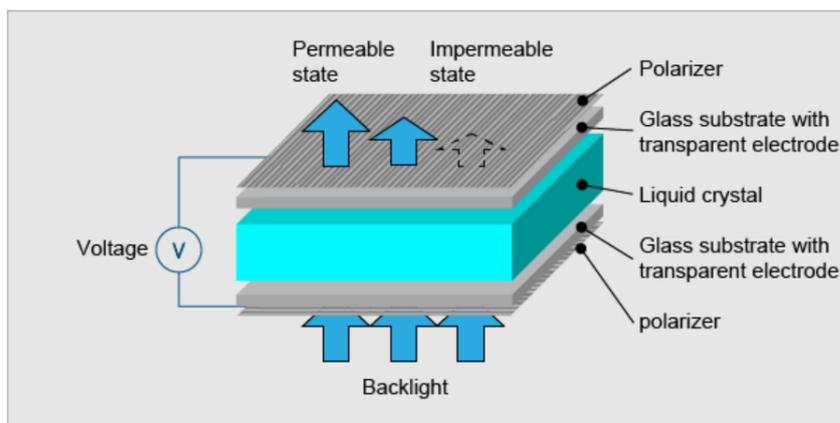


Figura 6: Princípio de funcionamento de um LCD [5].

A tela LCD TFT (*Thin Film Transistor*) é um tipo de tela sensível ao toque que é amplamente utilizada em *smartphones*, *tablets*, computadores, monitores e outros dispositivos eletrônicos [35].

Um TFT é um tipo de dispositivo semiconductor que serve como uma válvula de controle para fornecer uma voltagem apropriada em cristais líquidos para *sub pixels* individuais [34]. Na Figura 7 é possível observar como a tecnologia de tela sensível ao toque se dá por meio da inserção desses transistores, em cada *sub pixel* da tela que detectam a presença do dedo ou outra superfície condutora.

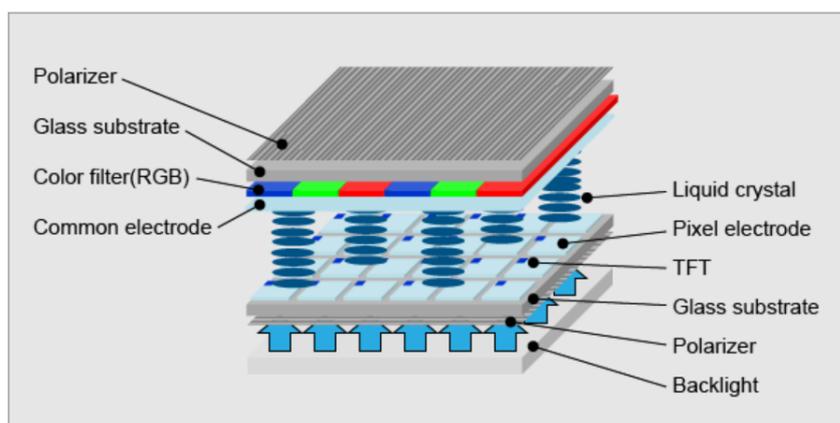


Figura 7: Estrutura do LCD TFT [5].

Os dados de toques detectados, são enviados para o processador do dispositivo que interpreta o gesto e executa a ação apropriada.

3 METODOLOGIA

O projeto desenvolvido tem como indicadores de sucesso uma maior autonomia nas transações dos produtos, ou seja, menor influência de pessoas no processo de negociação (pagamento e retirada do produto), uma melhor experiência do usuário/vendedor ao realizar uma transação e menor custo de produção do protótipo, visto que a solução é voltada para pequenos negócios, tudo isso comparado ao que se têm no mercado, hoje.

3.1 Open-Smart One R3 Air Easy-Plug

Baseado nos indicadores, o primeiro passo para melhora de autonomia e experiência do usuário foi o desenvolvimento de uma interface gráfica para apresentação dos produtos.

Dentre os diversos modelos de placas *open source* de baixo custo, a Open-Smart One R3 Air Easy-Plug foi escolhida por ter suporte para tela LCD TFT *Touch Screen* e conexão SPI para Micro SD. Por utilizar microcontrolador seu custo é bastante reduzido, comparado a outras opções como a placa Raspberry Pi com LCD (comumente utilizada para projetos que necessitam de interface gráfica).

O termo Easy-Plug está associado a modificação física de junção de todas as portas relevantes para a conexão do LCD TFT *Touch Screen*, facilitando o trabalho de conexão entre os dois componentes, essa modificação pode ser visualizada à direita da placa, na Figura 8.

Neste trabalho foi utilizada uma placa de desenvolvimento Open-smart One R3 Air com as especificações seguintes:

- Chip de driver USB: CH340G, compatível com Win, Linux e MAC OS;

- Microcontrolador: Atmel ATmega328P;
- Tensão de trabalho: 5V ou 3,3V;
- Tensão lógica IO: 5V ou 3,3V;
- Corrente de trabalho: 500mA (Max);
- Pinos de E/S digitais: 14 (dos quais 6 oferecem saída PWM);
- Memória *Flash*: 32 KB dos quais 2 KB usados pelo *bootloader*;
- RAM: 2KB;
- EEPROM: 1 KB;
- Velocidade do *clock*: 16 MHz.

A interface gráfica desenvolvida, mescla dois recursos diferentes que a placa disponibiliza, são eles: apresentação de imagens e leitura *Touch Screen*, desta forma, a partir de condicionais, foi possível elaborar um protótipo que conseguiu navegar entre imagens, ao pressionar áreas específicas da tela, simulando um aplicativo.

Para armazenar as imagens apresentadas na tela, foi necessário adquirir um módulo e cartão micro SD, que fazem parte do *kit* mostrado na Figura 8 e foram adquiridos na loja oficial da Open-Smart, no Aliexpress. As imagens estão disponíveis no apêndice A.

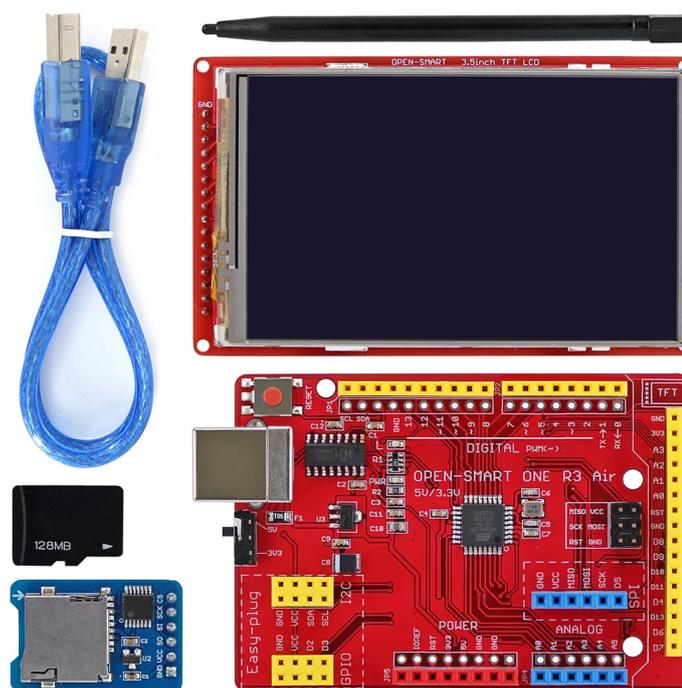


Figura 8: Kit Open-Smart One R3 Air utilizado no projeto [6].

3.2 Wemos D1 Mini

Como forma de melhoria dos indicadores de sucesso, o segundo passo foi diminuir a influência de pessoas no processo de transação. Dessa forma, foi essencial a escolha de uma placa com conexão Wi-Fi e suporte para bibliotecas de conexão HTTP/HTTPS.

Neste trabalho foi utilizada uma placa de desenvolvimento Wemos D1 Mini, exposta na Figura 9, com as especificações seguintes:

- Placa Wemos D1 Mini;
- Chip ESP8266 ESP-12F;
- 4 Mb de memória *Flash*;
- 11 Portas digitais que operam em nível lógico de 3,3V;
- Uma entrada analógica com resolução de 10 *bits*;
- Antena embutida;
- Entrada Micro USB com conversor USB/serial CH340G.

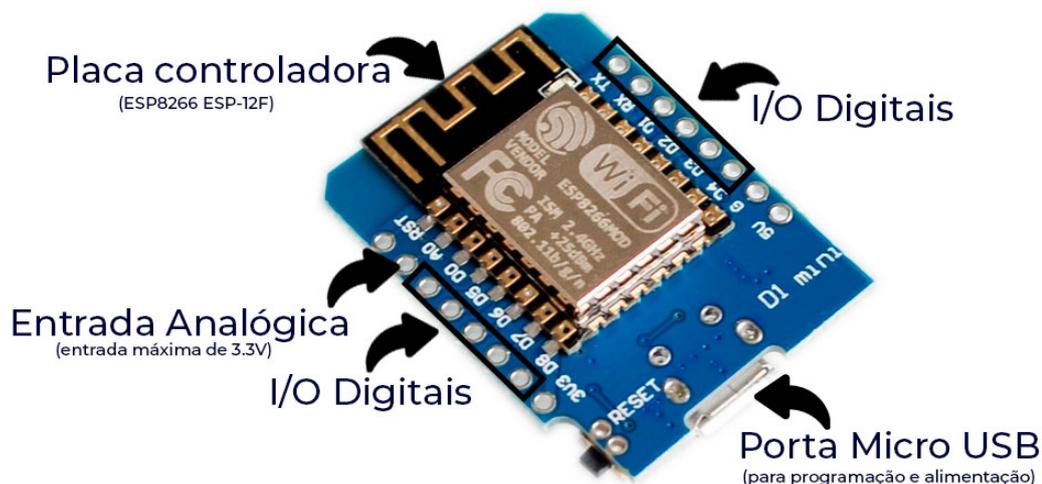


Figura 9: Placa de desenvolvimento Wemos D1 Mini utilizada no projeto [7].

O WeMos D1 Mini é uma placa de desenvolvimento compatível com o Arduino, além de ser uma placa de baixo custo e de tamanho compacto, tem como principal diferencial a conexão Wi-Fi (suporte às redes WiFi 802.11 b/g/n e criptografia WPA e WPA2). Sendo ideal para o desenvolvimento do que foi chamado de *gateway* de pagamento, tendo como função principal a realização de uma série de requisições HTTPS para a API do Asaas, afim de realizar transferências e consultas de saldo.

3.3 Asaas

O Asaas é uma fintech que oferece soluções para automação de negócios, autorizada pelo Banco Central, atuando como a 31ª instituição de pagamento do Brasil [36]. No projeto em questão foi utilizado a API de pagamentos do Asaas para realizar transferências via PIX e autenticar pagamentos.

A conexão com a API é feita via HTTPS e necessita de uma autenticação, dada por uma chave denominada de “access token”, inserida no *header* das requisições. Essa chave pode ser obtida nas configurações da conta digital e é única para cada conta.

Foi utilizado o método GET para obtenção do saldo da conta e há uma comparação constante entre as requisições, monitorando assim se houve entrada de capital.

O método POST também foi utilizado, apontando para um outro endereço HTTPS, sendo possível realizar transferências da conta Asaas para qualquer PIX especificado, enviando informações de valor a ser transferido e chave PIX, via *body* da requisição.

3.4 Protótipo

O protótipo é dividido em duas grandes partes: aplicação que executa na Open-Smart R3 Air, apelidada de *interface* gráfica, responsável por armazenar os dados de vendedores e exibir os produtos a venda, e a aplicação que executa na Wemos, apelidada de *gateway* de pagamento, responsável pela autenticação dos pagamentos e realização do encaminhamento de uma parte do valor para o vendedor. As duas partes se comunicam entre si, via Serial.

O protótipo da solução, nomeada de Honest.io, apresenta a ideia funcional do projeto, não se preocupando com as questões visuais, exposto na Figura 10.

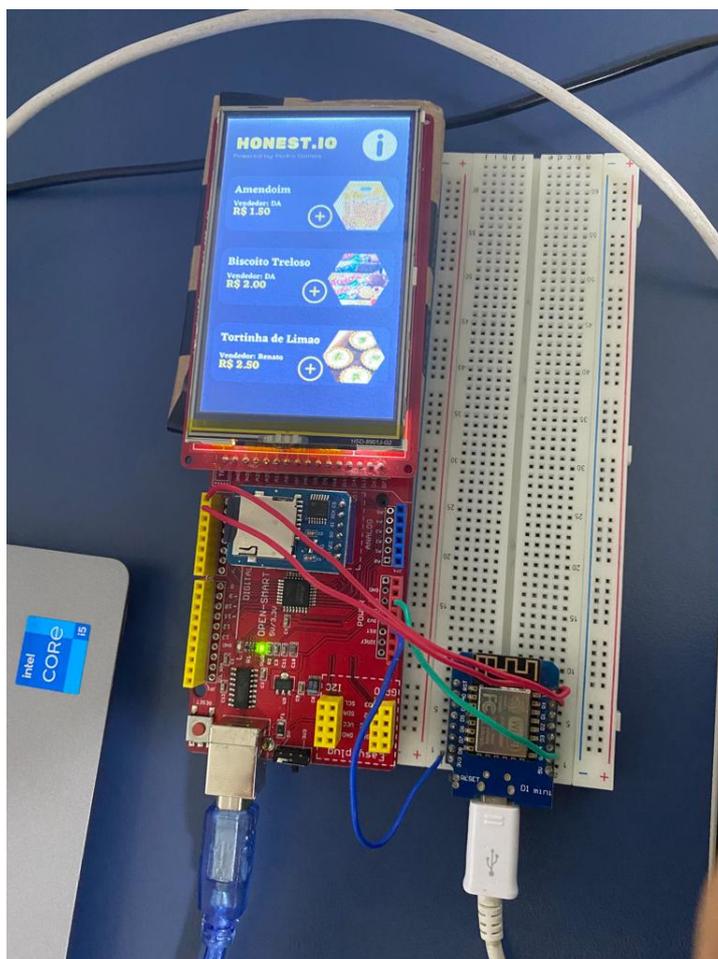


Figura 10: Protótipo de baixa fidelidade do Honest.io

Os códigos desenvolvidos para este projeto, bem como uma série de instruções para modificações dos parâmetros, são mantidos no GitHub [37] a fim de gerenciar as versões e possibilitar a contribuição de outros desenvolvedores no projeto.

3.4.1 Interface Gráfica

Consiste em uma placa OPEN-SMART R3, conectada a uma tela LCD TFT *Touch Screen* de 3.5 polegadas e um *shield* microSD, o sistema funciona como um aplicativo de vendas, que pode ser navegado através de simples toques na tela.

A tela principal contém os produtos cadastrados e após a escolha do produto, há um redirecionamento para a tela de pagamento.

A cada escolha de produto é enviado a chave pix do vendedor, o tipo de chave e o valor do produto, via Serial, para o *gateway* de pagamento.

Os dados enviados estão no formato “valor,chave(tipo_de_chave)”, isso concede a disponibilidade do vendedor de utilizar qualquer conta pessoal que tenha uma chave pix cadastrada, são aceitas chaves do tipo CPF, CNPJ, email, telefone e chave aleatória.

Após a identificação do pagamento (mensagem recebida, via Serial, do *gateway* para a *interface*) há um redirecionamento para uma tela de agradecimento e um retorno para a tela principal após alguns segundos.

As telas da *interface* gráfica foram produzidas utilizando o Canva e exportadas em PNG com 240x400 pixels, porém para utilização no projeto foi necessário a conversão em imagens BMP de 32 *bits*, nesse processo foi utilizado um conversor *on-line* [38], as telas estão disposta no apêndice A.

3.4.2 Gateway de Pagamento

O *gateway* é constituído por uma Wemos D1 Mini conectada ao Wi-Fi, esse sistema realiza seguidas requisições de saldo utilizando o método GET de HTTPS. O sistema compara o saldo recebido nas duas últimas requisições tornando possível a identificação de pagamentos.

Como visto anteriormente, ao acessar a tela de pagamento na *interface* gráfica, é enviado, via Serial da *interface* para o *gateway*, uma *string* especificando os dados do produto e vendedor, esses dados são tratados no *gateway* e os valores a serem recebidos

são atualizados a cada produto escolhido na *interface*.

Após a identificação do pagamento, é executado, automaticamente, uma requisição utilizando o método POST de HTTPS, responsável por enviar, via PIX, 90% do valor pago para o vendedor cadastrado e enviado.

4 RESULTADOS

4.1 Primeira sessão de testes

A primeira sessão de testes teve como objetivo avaliar o desempenho do sistema diante de simulações de compra, utilizando vários tipos de chaves PIX e redes Wi-Fi diferentes.

Os testes foram realizados em um ambiente controlado e por pessoas que conheciam o funcionamento da plataforma.

O protótipo conseguiu desempenhar bem o monitoramento de pagamento e a transferência da porcentagem pré estabelecida no código (após o pagamento).

Um dos pontos positivos que foram levantados é o fato do vendedor receber via PIX e não ficar restrito a um banco específico, podendo usar qualquer conta pessoal que tenha uma chave PIX cadastrada.

Como ponto de melhoria, foi observado que ao perder a conexão Wi-Fi o dispositivo não apresenta nenhum tipo de sinalização, o *touch screen* continua funcionando normalmente, porém sem conseguir validar os pagamentos e realizar as transferências automáticas.

Um outro ponto observado, foi a demora na transição entre as imagens do visor, que pôde ser contornado, aumentando a quantidade de memória utilizada na escrita dos pixels.

4.2 Segunda sessão de testes

Antes de iniciar a segunda sessão de testes, foram realizadas algumas modificações na *interface* do usuário, colocando o protótipo em um *case* de papelão e algumas instruções de uso coladas ao lado do visor.

Essa sessão de testes teve como objetivo realizar uma prova de campo, sendo possível analisar o comportamento do protótipo frente ao estresse do uso contínuo e diário, durante uma semana.

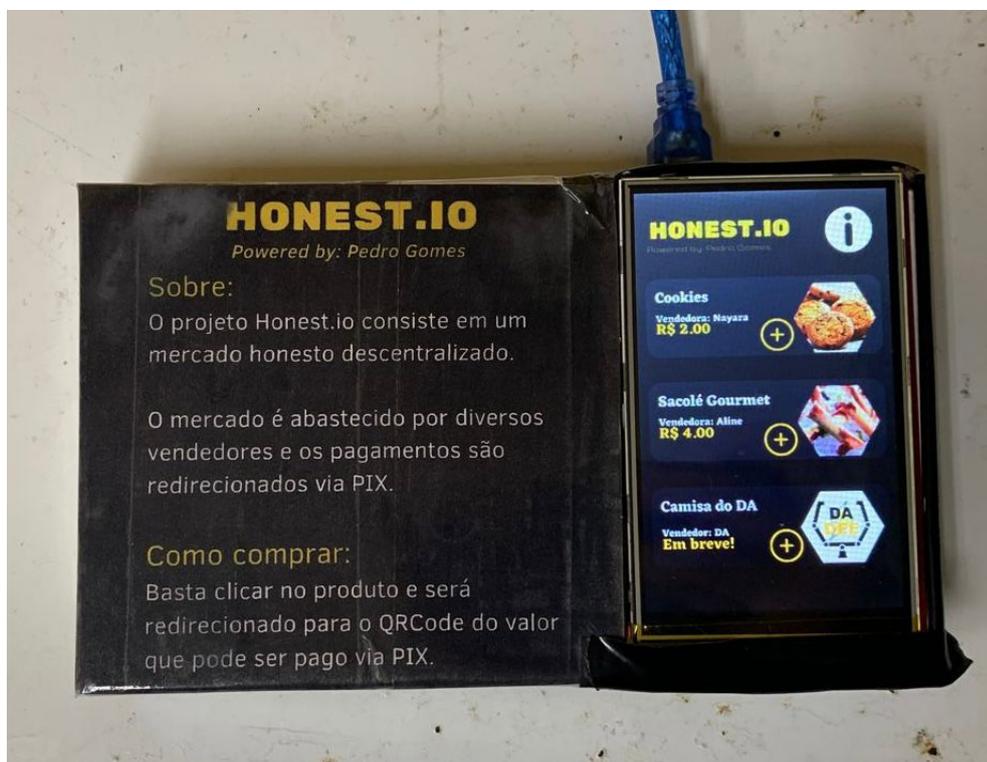


Figura 11: Protótipo de alta fidelidade do Honest.io

O protótipo foi implementado no Diretório Acadêmico de Engenharia Elétrica (DAEE) da UFPE, foram cadastrados 3 produtos de vendedores distintos e fixado uma taxa de 10% sobre o valor da venda, essa taxa é destinada ao DA (produtos vendidos pelo DA não são taxados). Na Tabela 2 é possível observar os produtos cadastrados para a segunda sessão de testes, bem como os valores, taxas e vendedores.

Tabela 2: Produtos negociados na segunda sessão de testes

Produto	Vendedor (a)	Valor Bruto	Taxa	Valor do DA	Valor do vendedor
Cookies	Nayara	R\$ 2,00	10%	R\$ 0,20	R\$ 1,80
Sacolé Gourmet	Aline	R\$ 4,00	10%	R\$ 0,40	R\$ 3,60
Camisa DA	DA	R\$ 35,00	0%	R\$ 0,00	R\$ 35,00

As vendas foram monitoradas durante uma semana, sendo possível ter uma amostra do cotidiano de um pequeno negócio deste tipo. Os dados semanais de quantidade negociada e valores recebidos podem ser observados na Tabela 3.

Tabela 3: Resultado das negociações em um período semanal

Produto	Qnt. Vendas	Total Recebido	Recebido DA	Recebido Vendedor
Cookies	23	R\$ 46,00	R\$ 4,60	R\$ 41,40
Sacolé Gourmet	13	R\$ 52,00	R\$ 5,20	R\$ 46,80
Camisa DA	0	R\$ 00,00	R\$ 00,00	R\$ 00,00
Total	36	R\$ 98,00	R\$ 9,80	R\$ 88,20

A taxa fixada é de 10%, visto que a ideia principal é realmente ajudar os estudantes. Por outro lado, não há nenhum tipo de valor recorrente de manutenção do protótipo (o funcionamento da API é gratuita para as funcionalidades utilizadas e a internet e energia são fornecidas pela universidade). O objetivo principal do dinheiro, coletado pelo DA, é ser reinvestido nos próprios alunos, melhorando a infraestrutura do DA.

4.3 Terceira sessão de testes

A solução apresentada neste projeto, foi convidada para ser exposta no Arduino-Day, evento mundial dedicado à celebração da plataforma Arduino e sua comunidade, promovido em Recife-PE, pelo Cesar School, no dia 26 de março de 2023 [39].

Este evento foi uma grande oportunidade de realização de mais uma sessão de testes, visando o grande volume de vendas que poderiam ser feitas em um único dia. Desta forma, o Honest.io foi reprogramado para vender produtos de baixo valor, por um valor de revenda normal, os produtos escolhidos foram paçoca, pirulito e pastilhas.

O projeto foi exposto durante o dia, no Cesar School, como visto na Figura 12, obtendo como resultado um quantitativo de 62 transações de 32 clientes diferentes. Foi observado que uma boa parte dos clientes conseguiu utilizar o Honest.io sem nenhum tipo de ajuda, apenas seguindo as instruções da tela e não houve nenhum produto subtraído, mesmo o projeto funcionando sem nenhum tipo de acompanhamento, em alguns momentos.

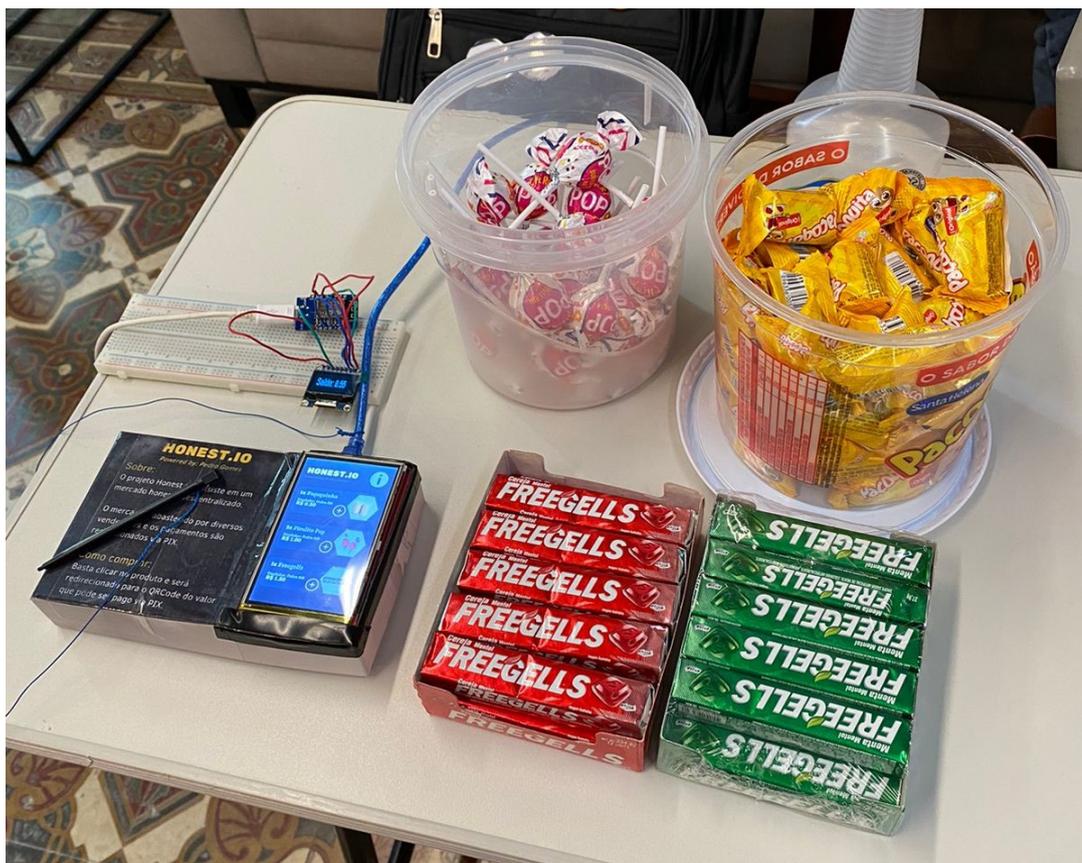


Figura 12: Honest.io exposto no ArduinoDay

Após uma quantidade expressiva de 50 transações, houve cobranças, por parte do Asaas, de um valor relativamente alto para o funcionamento normal da solução apresentada. Foi cobrado um valor de R\$ 0,99 por PIX recebido, deixando inviável a venda de produtos muito baratos. Uma forma de contornar isso seria a inserção de outras contas Asaas, no processo, e um revezamento dos PIX recebidos, tentando não extrapolar esses valores de transações limites, já que até esse limite a API funciona de forma gratuita.

A taxa cobrada é um valor relativamente baixo relativo ao uso da API, porém nossas transações giravam em torno de R\$ 1,00, sendo cobrado 100% do valor de PIX recebido, desabilitando totalmente o projeto para valores baixos.

Como sugestão, poderia ser colocado produtos a partir de R\$ 10,00, cobrando uma taxa de 20% ao vendedor, sendo possível absorver a taxa da API e garantir o funcionamento, mesmo com altas quantidades de transações.

5 CONSIDERAÇÕES FINAIS

5.1 Conclusões

A tecnologia junto à pandemia mudou em diversos aspectos as relações comerciais do mundo, fazendo crescer o mercado de *marketplace* e difundir o conceito de mercado autônomo. Já na universidade, há uma comercialização muito forte de diversos produtos por parte dos alunos, buscando um auxílio financeiro para custeio de sua passagem pela faculdade.

Nesse contexto, este trabalho visou desenvolver um sistema físico de gerenciamento de vendas de produtos de diversos alunos diferentes, criando um ecossistema ao seu redor, baseado nos conceitos de mercado autônomo e *marketplace*.

Em relação ao preço total do projeto, o Kit Open-Smart One R3 [6] custa em média R\$ 100,00, uma Wemos D1 mini custa em torno de R\$ 30,00. Portanto, o custo final de produção do projeto é de R\$ 130,00, sendo bem abaixo dos preços de máquinas de cartão inteligentes do mercado, que custam cerca de R\$ 450,00 e normalmente só funcionam junto ao pagamento de uma mensalidade.

É importante ressaltar a utilização do protocolo HTTPS para comunicação, via internet, a partir da ESP8266 e como isso abre diversas portas para imaginação.

5.2 Sugestões para trabalhos futuros

Tendo em vista uma maior experiência dos usuários e maior disponibilidade do sistema, algumas sugestões foram dadas para um aperfeiçoamento deste projeto.

- Aumento na quantidade de produtos e possibilidade de compra de vários produtos de uma só vez;
- Uma parceria com o Asaas, a fim de reduzir as taxas após a quantidade limite de PIX gratuitos recebidos ou mudança de API para uma API com taxas percentuais;
- Implementação de um sistema via *web*, podendo alterar os produtos diretamente por lá;
- Transferência de variáveis para o SD, não precisando carregar o código ao realizar mudanças de preço e informações de Wi-Fi;
- Taxas variadas de acordo com o produto, possibilitando a venda de produtos pelo mantenedor da solução (com taxa zero);
- Criação de um *layout* para a placa, reduzindo o tamanho, bem como um *case* 3D para melhoria da parte visual do projeto;
- Implementação de um processo de validação constante na conexão com a internet e aviso quando o sistema estiver indisponível;
- Incremento de memória RAM, a fim de aumentar a velocidade de atualização das telas;
- Implementação de um processo automatizado de gerar os arquivos BMP, só precisando alterar as telas no Canvas.

REFERÊNCIAS

- 1 HTTP Status Codes: List of HTTP Response Status Codes. Disponível em: <<https://www.infidigit.com/blog/http-status-codes/>>.
- 2 KEY differences Between TLS 1.2 and TLS 1.3. Disponível em: <<https://www.a10networks.com/glossary/key-differences-between-tls-1-2-and-tls-1-3/>>.
- 3 COMUNICAÇÃO serial Arduino via protocolo UART. Disponível em: <<https://makerhero.com/blog/comunicacao-serial-arduino-via-protocolo-uart/>>.
- 4 COMUNICAÇÃO SPI em Linux. Disponível em: <<https://embarcados.com.br/wp-content/uploads/2015/04/spi-em-linux.png>>.
- 5 TECHNOLOGY: LCD Basics | Japan Display Inc. Disponível em: <<https://www.j-display.com/english/technology/lcdbasic.html>>.
- 6 KIT de exibição 400x240 tela touch + easy-plug + módulo microsd. Disponível em: <https://pt.aliexpress.com/item/32964204802.html?spm=a2g0o.order_detail.order_detail_item.3.87464c7fSm8TUN&gatewayAdapt=glo2bra>.
- 7 WEMOS D1 Mini. Disponível em: <https://blogmasterwalkershop.com.br/wp-content/uploads/2021/05/img01_especificacoes_conheca_o_wemos_mini_d1_pequena_no_tamanho_e_imensa_nas_funcionalidades_esp8266_esp01_esp12_esp12e_automacao_robotica_arduino.jpg>.
- 8 RELATÓRIO E-commerce no Brasil Abril/2021. Disponível em: <https://www.comecomm.com.br/wp-content/uploads/2021/04/Abr_21-Mar-Relato%CC%81rio-E-commerce-no-Brasil-1.pdf>.
- 9 BUKMAN, G. Desenvolvimento de um protótipo de baixo custo para segurança residencial. 45 f. Monografia (Graduação) — Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2016. Disponível em: <<http://www.monografias.ufop.br/handle/35400000/280>>.
- 10 RUBIO, B. U. et al. Alert system to control water consumption for domestic service. In: . [S.l.: s.n.], 2022. p. 86–91.
- 11 O crescimento dos marketplaces em 2021 - E-commerce Brasil. Disponível em: <<https://www.ecommercebrasil.com.br/artigos/o-crescimento-dos-marketplaces-em-2021>>.
- 12 LOCAWEB - Marketplace: O que é e como pode transformar o seu negócio? Disponível em: <<https://www.locaweb.com.br/blog/temas/primeiros-passos/o-que-e-marketplace/>>.

- 13 FECOMERCIO SP - Transitório ou permanente: A pandemia e o novo comportamento do consumidor. Disponível em: <https://lab.fecomercio.com.br/wp-content/uploads/2020/09/ebook-transitorio_APROV_final.pdf>.
- 14 FOLHA de Londrina - Mercados autônomos entram em fase de consolidação. Disponível em: <<https://www.folhadelondrina.com.br/economia/mercados-autonomos-entram-em-fase-de-consolidacao-3165963e.html#:~:text=Em%202021%2C%20a%20startup%20registrou,a%20mil%20em%20um%20ano.>>
- 15 MINHA Quitandinha - Quem Somos. Disponível em: <<https://www.minhaquitandinha.com.br/quem-somos/>>.
- 16 SUPER Domínios - Por que os entusiastas da tecnologia adoram usar o domínio .IO. Disponível em: <<https://superdominios.org/por-que-os-entusiastas-da-tecnologia-adoram-usar-o-dominio-io/>>.
- 17 ROCKCONTENT - Entenda o que é HTTP. Disponível em: <<https://rockcontent.com/br/blog/http/>>.
- 18 MOZILLA Developer - HTTP Overview. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>>.
- 19 W3SCHOOLS - HTTP Methods. Disponível em: <https://www.w3schools.com/tags/ref_httpmethods.asp>.
- 20 SSL 3.0 Protocol Vulnerability and POODLE Attack. Disponível em: <<https://www.us-cert.gov/ncas/alerts/TA14-290A>>.
- 21 SECURITY UFRJ - Servidores que utilizam ssl, vulnerabilidades e soluções. Disponível em: <<https://www.security.ufrj.br/tutoriais/servidores-que-utilizam-ssl-vulnerabilidades-e-solucoes/>>.
- 22 KING Host - SSL TLS Entenda as tecnologias de segurança da rede. Disponível em: <<https://king.host/blog/servicos-essenciais/ssl-tls-entenda-as-tecnologias-de-seguranca-da-rede/>>.
- 23 SERASA - Certificado Digital - SSL. Disponível em: <<https://serasa.certificadodigital.com.br/blog/mercado/o-que-e-o-certificado-digital-ssl-e-como-ele-pode-protger-informacoes/>>.
- 24 IBM - TLS Overview SSL Handshake. Disponível em: <<https://www.ibm.com/docs/pt-br/ibm-mq/8.0?topic=tls-overview-ssl-handshake>>.
- 25 MICROSOFT - Design da API Web RESTful. Disponível em: <<https://learn.microsoft.com/pt-br/azure/architecture/best-practices/api-design>>.
- 26 API RESTful Explanation - TechTarget. Disponível em: <<https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>>.
- 27 ARDUINO - Introduction Guide. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>.

28 TUTORIALSPPOINT - Arduino board description. Disponível em: <https://www.tutorialspoint.com/arduino/arduino_board_description.htm>.

29 VICTORVISION - Interface UART. Disponível em: <<https://victorvision.com.br/blog/interface-uart/#:~:text=A%20interface%20UART%20%C3%A9%20uma%20tecnologia%20que%20permite,apenas%20dois%20fios%3A%20muito%20mais%20praticidade%20e%20organiza%C3%A7%C3%A3o.>>

30 ANALOG - UART a hardware communication protocol. Disponível em: <<https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>>.

31 ARDUINO - Serial Communication. Disponível em: <<https://www.arduino.cc/reference/pt/language/functions/communication/serial/>>.

32 COMUNICAÇÃO SPI (Serial Peripheral Interface). Disponível em: <<https://embarcados.com.br/spi-parte-1/>>.

33 ARDUINO Serial Peripheral Interface (SPI). Disponível em: <<https://docs.arduino.cc/learn/communication/spi>>.

34 HOW does TFT LCD Display Work. Disponível em: <<https://www.orientdisplay.com/knowledge-base/tft-basics/how-does-tft-displays-work/>>.

35 WHAT is LCD (Liquid Crystal Display)? Disponível em: <<https://www.techtarget.com/whatis/definition/LCD-liquid-crystal-display>>.

36 SOBRE o Asaas: conheça quem somos e nossa trajetória. Disponível em: <<https://www.asaas.com/sobre-nos>>.

37 REPOSITÓRIO Github - Honest.io - Pedro Gomes. Disponível em: <<https://github.com/peddrogomes/honest.io>>.

38 BMP Converter - convert PNG to BMP. Disponível em: <<https://online-converting.com/image/convert2bmp/>>.

39 ARDUINODAY - Recife, 2023, Sympla. Disponível em: <<https://www.sympla.com.br/evento/arduino-day-recife-2023/1913069>>.

APÊNDICE A

A.1 Imagens apresentadas na tela LCD



Figura 13: Tela principal da interface gráfica

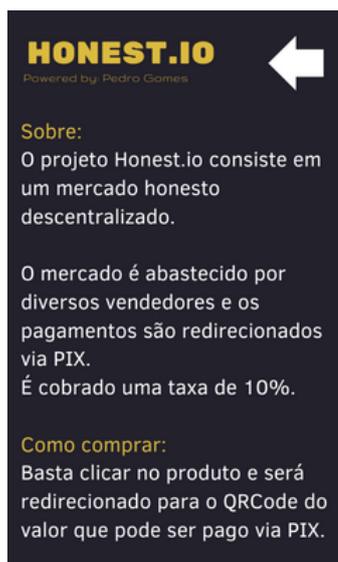


Figura 14: Tela de explicação do sistema

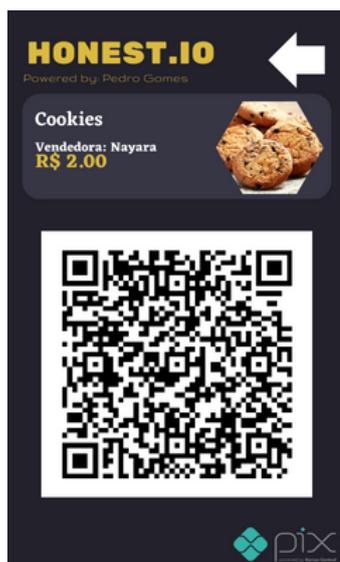


Figura 15: Tela de QRCode de pagamento



Figura 16: Tela de agradecimento