

Relatório do Trabalho II de Redes sem Fios: Smart Box

Alisson Pereira Ferreira¹
Emir Bráz de Araújo Marques Júnior¹
João Pedro Tavares Santos¹
Monique Rosa Moraes¹
Nícolas André Baümle¹
Régis Nyland Bloemer¹
Rodrigo Guedes de Souza¹

¹Universidade Federal de Santa Catarina (UFSC) – Campus Araranguá
Caixa Postal 88.905-120 – Araranguá – SC – Brasil

Abstract. *This work describes the development of a monitoring interface for self-service points of sale located in the entrance hall of UFSC Araranguá Campus, using a box to store the products, end-of-stroke sensors to verify the box opening, weight sensors to detect product removal, and an ESP32-CAM module for buyer control and identification. The transmission of data collected by the components to the cloud will be done through Firebase, allowing access through any device with Internet access.*

Resumo. *Este trabalho descreve o desenvolvimento de uma interface de monitoramento para pontos de venda self-service presentes no hall de entrada do Campus UFSC Araranguá, utilizando uma caixa, para armazenar os produtos, um sensor de fim de curso, para verificar a abertura da caixa, um sensor de peso conectado a um ESP32, para detectar a retirada de produtos e enviar um controle de estoque à nuvem, e um módulo ESP32-CAM, para a identificação dos compradores. A transmissão dos dados coletados pelos componentes para a nuvem será realizada pelo Firebase, permitindo o acesso através de um dispositivo qualquer conectado à Internet.*

1. Introdução

A utilização de redes sem fios vem crescendo cada vez mais, trazendo novas tecnologias e possibilitando múltiplas integrações entre os mais diversos setores, do comércio à automação. Dentre os vários padrões e protocolos, as redes móveis fornecem maior liberdade, conectividade e aplicações robustas, como a Internet das Coisas (Internet of Things - IoT). O 4G é a quarta geração de redes móveis e é baseado totalmente em IP, o que significa que todos os serviços, como voz, dados e multimídia, são transmitidos pela Internet. Além disso, o 4G introduziu o conceito de VoLTE (Voice over LTE), permitindo que chamadas de voz sejam feitas pela rede LTE (Long Term Evolution) ao invés de redes legadas separadas, resultando em chamadas de voz de alta qualidade e recursos avançados, como chamadas em HD.

Dentre as diversas tecnologias existentes, o 4G foi escolhido para este trabalho devido a maior estabilidade em comparação às redes Wi-Fi disponíveis no campus, como a "eduroam". Além disso, por questões de segurança, o tráfego nas redes da UFSC é

privado e controlado, e não permite implementações externas ao sistema, como a que será desenvolvida neste projeto.

De modo geral, como já descrito no resumo, esse trabalho busca desenvolver uma interface autônoma de monitoramento para pontos de venda self-service presentes no hall de entrada do Campus UFSC Araranguá, utilizando sensores de fim de curso e de peso, módulo ESP32-CAM, ESP32 e a ferramenta Firebase para o envio dos dados à nuvem.

O projeto foi escolhido porque muitos estudantes aproveitam o espaço do hall para vender produtos, como alimentos e bebidas, e obter uma fonte de renda alternativa. Porém, como a rotina acadêmica é complexa e conciliar os horários e permanecer continuamente no espaço não é possível, então optar pelo autoatendimento foi a solução encontrada pelos discentes. A ideia é viável, mas controlar as compras acaba sendo um desafio considerável, pois não há monitoramento. Este projeto busca resolver esse problema.

2. Lista de Itens

A tabela abaixo apresenta os itens que serão utilizados para a implementação do projeto e suas quantidades. As imagens, por sua vez, facilitam a compreensão e a diferenciação dos itens.

Item	Quantidade
Módulo ESP32-CAM com Câmera OV2640 2MP	1
ESP32	1
Módulo HX711	1
Sensor/célula de carga de 5 kg	1
Sensor de fim de curso KW12-3	1

Tabela 1. Itens utilizados e quantidades.

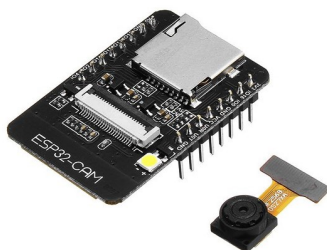


Figura 1. Módulo ESP32-CAM com Câmera OV2640 2MP.

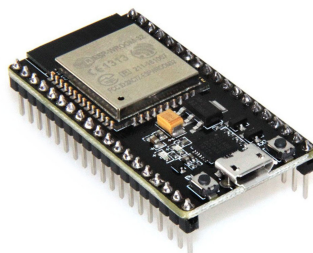


Figura 2. ESP32.

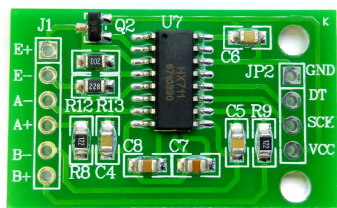


Figura 3. Módulo HX711.

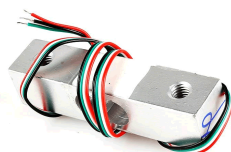


Figura 4. Sensor de carga de 5 kg.



Figura 5. Sensor de fim de curso KW12-3.

3. Lista de Riscos

Para "Probabilidade" e "Severidade", há uma classificação por níveis e cores, como apresentado a seguir. Para "Categoria", obtida pela *Probabilidade × Severidade*, e definida no intervalo [1,16], a mesma classificação de cores é aplicada, atentando-se à divisão de intervalos de largura 4, ou seja, Baixo [1,4], Moderado [5,8], Elevado [9,12] e Grave [13, 16].

Nível	Classificação
1	Baixo
2	Moderado
3	Elevado
4	Grave

Tabela 2. Classificação dos riscos.

Número	Risco	Probabilidade (1-4)	Severidade (1-4)	Categoria (1-16)
1	Cronograma apertado comprometer a qualidade do protótipo	4	4	16
2	Atraso na entrega da câmera e sensores	2	3	6
3	Falha dos equipamentos e não junção das partes	1	4	4
4	Faltas de integrantes em aula	4	3	12
5	Atraso no cronograma semanal	2	3	6
6	Delay da câmera ou falta de comprometimento dos membros da equipe	1	3	3
7	Problemas técnicos	4	3	12
8	Dificuldade de integração das partes do trabalho	2	2	4
9	Não validar a Smart Box	1	2	2
10	Feriados em aula	4	2	8
11	Não conseguir criar um banco de dados a tempo	2	1	2
12	Relatório incompleto	1	3	3

Tabela 4. Lista de Riscos.

Risco	Atenuações
1	Revisar o cronograma e identificar tarefas prioritárias.
2	Comunicar imediatamente o fornecedor sobre o atraso e buscar alternativas.
3	Realizar verificações regulares dos equipamentos e partes do projeto.
4	Comunicar claramente a importância da participação de todos os membros e estabelecer um compromisso de presença nas aulas.
5	Reavaliar as tarefas prioritárias e redistribuir recursos para recuperar o tempo perdido.
6	Estabelecer prazos claros e responsabilidades individuais.
7	Realizar manutenção preventiva dos equipamentos e conversas com pessoas que dominam o assunto.
8	Estabelecer uma comunicação eficiente entre os membros da equipe e promover a colaboração.
9	Realizar testes e validações periódicas.
10	Antecipar os feriados em aula e ajustar o cronograma de acordo.
11	Avaliar a viabilidade de utilizar uma solução pré-existente ou simplificada para o banco de dados.
12	Revisar o escopo do relatório e priorizar as seções mais importantes.

Tabela 3. Atenuações dos Riscos.

A aplicação [Notion](#) foi utilizada para o gerenciamento do projeto, permitindo que a equipe acompanhe o andamento de suas tarefas e dados importantes para o desenvolvimento.

4. Estado da Arte

Para o estado da arte, a base IEEE Xplore foi utilizada, empregando a seguinte *string* de busca: *ESP32 AND 4G AND Sensor*. Os resultados foram filtrados para localizar os estudos dos 5 últimos anos, de 2019 a 2023, obtendo-se, assim, 3 artigos de conferência.

O primeiro estudo, *Renewable Energy in Smart Grid: Photovoltaic Power Monitoring System Based on Machine Learning Using an Open-Source IoT Platform*, por [\[Yang et al. 2022\]](#), propôs um sistema de monitoramento fotovoltaico baseado em Internet das Coisas (IoT) utilizando a placa Raspberry Pi como servidor e o ESP32 como cliente, comunicando-se por meio dos protocolos MQTT e HTTP. O sistema é dividido em duas etapas: na primeira etapa são realizadas medições de potência dos painéis solares a cada 5 minutos. A tensão e a corrente são medidas pelo sensor ACS712, enquanto a temperatura dos painéis solares é medida por um sensor DHT. Essas medidas são exibidas na plataforma Node-RED e armazenadas em um banco de dados SQLite. Posteriormente, um cartão Raspberry Pi registra esses dados em tempo real por meio de conexão Wi-Fi ou Ethernet. Já na segunda etapa, com base nos dados coletados anteriormente, é aplicado um algoritmo de *machine learning* para prever a energia fornecida pelos painéis. O Node-RED é a

interface utilizada para aplicar o algoritmo e permite monitorar as medições em tempo real por meio de um PC (Wi-Fi) e um smartphone (4G). Como base nos resultados obtidos, um sistema inteligente foi desenvolvido, possibilitando um gerenciamento mais eficiente, automatizado e flexível dos recursos existentes.

O segundo artigo, *Machine Learning-based Smart Irrigation Monitoring System for Agriculture Applications Using Free and Low-Cost IoT Platform*, por [Youness et al. 2022], desenvolveu uma solução para o Sistema de Monitoramento de Irrigação Inteligente da Internet das Coisas com base em Inteligência Artificial. A solução envolve a comunicação entre um Raspberry Pi3 e vários clientes ESP32 utilizando os protocolos MQTT e HTTP. Na primeira parte, é realizada a medição da umidade do solo em várias áreas do terreno. Na segunda parte, são utilizados painéis fotovoltaicos como fonte de energia. A potência e a energia dos painéis solares são medidas a cada 5 minutos por meio de um sensor ACS712, e os dados são exibidos na plataforma Node-RED e armazenados em um banco de dados SQLite para projeções de água e umidade necessárias para o solo. Na última parte, é utilizado um algoritmo de *machine learning* para prever a proporção da água reservada, permitindo o acionamento da válvula de forma adequada. O Node-RED é a interface mais adequada para a aplicação do algoritmo e permite o monitoramento em tempo real por meio de um PC (conexão local) e um smartphone (conexão 4G).

Por fim, o artigo *Mudslide Disaster Monitoring and Early Warning System Based on ESP32*, por [Hakam et al. 2023], descreve o desenvolvimento de um sistema de monitoramento e alerta utilizando um ESP32 aliado a sensores de chuva, umidade, infrassom e um fractômetro (instrumento de medição mecânica para determinar valores característicos de resistência à flexão e compressão) para obter os dados locais. Um módulo 4G foi empregado para a transmissão dos dados para o servidor em nuvem para processamento e avaliação do risco de desastres. Após o processamento, os resultados são exibidos no aplicativo WeChat.

Todos os trabalhos encontrados tratam de condições semelhantes às abordadas neste projeto, utilizando o ESP32, sensores e a rede móvel 4G para transmissão dos dados. A busca não retornou muitos resultados, mas foi capaz de demonstrar a grande variedade de aplicações que o ESP32 oferece, mesmo sendo de baixo custo. O artigo de [Hakam et al. 2023], que aborda o monitoramento de desastres, é o mais interessante entre os 3 encontrados, visto que fornece referência para a utilização de sensores e instrumentos pouco utilizados, como o infrassom e o fractômetro, respectivamente.

5. Modelagem do Sistema

O sistema de monitoramento de vendas no hall da UFSC Araranguá consiste em uma caixa equipada com diversos componentes para acompanhar e registrar as transações de venda.

A caixa, adaptada para acomodar os produtos vendidos, é equipada com um sensor de fim de curso, que detecta a abertura e fechamento da tampa, e um sensor de peso, localizado no fundo da caixa, que é utilizado para indicar a retirada de produtos. Quando um produto é retirado da caixa, o sensor de peso registra a diminuição da carga e envia um sinal para o sistema, que processa os dados e captura uma foto do cliente.

O módulo ESP32-CAM é responsável por alimentar todos os sensores presentes no sistema, capturar os dados, incluindo as imagens, processá-los condicionalmente e

enviá-los ao Firebase, um serviço que permite que os dados do sistema sejam armazenados e acessados de forma rápida, segura e remota.



Figura 6. Confeção da Smart Box.

6. Resultados e Discussões

O desenvolvimento do trabalho demandou grande empenho da equipe. Inicialmente, a ativação da câmera foi simples, mas a transmissão para a nuvem e o tratamento condicional da captura — ou seja, a captura da foto após a abertura da caixa, indicada pelo sensor de fim de curso — apresentaram complicações significativas.

Após algumas semanas de investigação e pesquisa, uma biblioteca foi encontrada, permitindo que as instabilidades do sensor de fim de curso fossem tratadas e seu funcionamento aprimorado, o que acelerou a agenda do projeto. Quanto à transmissão para a nuvem, o maior obstáculo foi a sobrescrita das imagens, mesmo após a alteração do código e a aplicação de lógicas distintas. O problema foi resolvido semanas depois a partir de uma lógica específica, que aplicou um *timestamp* para identificar cada imagem capturada pelo módulo ESP32-CAM.

Por outro lado, o processo de implementação do sensor de peso ocorreu, basicamente, em duas fases. Na primeira fase, foi empregado o o módulo ESP32-CAM em conjunto com o módulo HX711 e o sensor de peso de 50 kg disponível no laboratório. Durante esta fase, ocorreu a alteração e melhoria de um código, encontrado por um dos integrantes do grupo, para atender às necessidades e especificações do projeto. No entanto, a etapa resultou em inconsistências nas medições fornecidas pelo sensor, como a apresentação de valores de peso negativos.

Após inúmeras tentativas de depuração, surgiu a hipótese de que a origem do problema fosse o sensor de peso, o que levou à decisão de trocá-lo por outro sensor com as mesmas configurações e utilizar um ES32 à parte para configurá-lo. Porém, as inconsistências persistiram. Uma possível solução surgiu com o empréstimo de um sensor de peso mais refinado e preciso cedido pelo professor Jim Lau. A implementação deste novo sensor seguiu uma metodologia semelhante a do anterior, no entanto, foram enfrentadas dificuldades na conexão entre o sensor e o ESP32, mesmo com a utilização

de bibliotecas pertinentes e consulta a materiais especializados, como o fornecido pelo professor Jim Lau.

A solução adotada, considerando o pouco tempo restante para a finalização do projeto, consistiu no desenvolvimento de um servidor, utilizando-se o *framework* Flask, para estabelecer a comunicação e o envio das medições. Devido à incapacidade de coletas reais, optou-se pelo envio de dados aleatórios como solução temporária ao impasse.

Apesar do atraso no desenvolvimento do projeto e do relatório gerado pelas dificuldades e limitações encontradas, a equipe soube lidar com as problemáticas, participando ativamente na busca e implementação de soluções em todos os níveis. Como resultado, conseguiu-se alcançar o principal e mais difícil objetivo do trabalho: utilizar o módulo ESP32-CAM para capturar e transmitir imagens à nuvem a partir da leitura de um sensor de fim de curso. Além disso, foi possível construir um protótipo detalhado do sistema, fornecendo à equipe diversos incentivos para a continuidade do projeto e para a criação de novos trabalhos e soluções.

6.1. Links importantes

A Tabela 5 apresenta os repositórios dos códigos utilizados para a elaboração do projeto.

Módulo ESP32-CAM + Sensor de fim de curso
ESP32 + Sensores de peso

Tabela 5. Links dos repositórios.

O segundo repositório apresenta todos os códigos implementados com os sensores de peso, desde o uso inicial de um sensor de 50 kg até a transição para um sensor mais preciso de 5 kg, disponibilizado pelo professor Jim Lau. Além disso, este repositório também inclui todos os recursos desenvolvidos para a criação do servidor a partir do *framework* Flask.

7. Conclusão

O presente trabalho teve como objetivo o desenvolvimento de uma interface de monitoramento para pontos de venda self-service no hall de entrada do Campus UFSC Aranguá, aplicando-se técnicas de redes sem fio, sensores e microcontroladores. Apesar dos percalços encontrados, a equipe demonstrou compromisso e conhecimento na elaboração do projeto.

Os resultados alcançados, apesar das modificações exigidas para a aplicação do sensor de peso, demonstram a funcionalidade e aplicabilidade da ideia original no contexto universitário. Sem dúvida, esses resultados fornecem subsídio técnico e programático para a continuidade do projeto e o desenvolvimento de novas aplicações.

Para trabalhos futuros, sugere-se a exploração de diferentes modelos de sensores de peso e a aplicação de outras bibliotecas ou *frameworks* que facilitem a implementação e melhorem a qualidade das medições. Além disso, seria relevante um estudo de casa para avaliar as funcionalidades do sistema, buscando, talvez, inserir diferentes tipos de sensores e regras de negócio que otimizem a operação da *Smart Box*.

De modo geral, os objetivos iniciais do projeto foram alcançados, e a experiência do desenvolvimento proporcionou aprendizados valiosos à equipe, reiterando a importância do compromisso, da responsabilidade e da aplicação de múltiplos conhecimentos e habilidades.

Referências

- [Delaney 2018] Delaney, J. (2018). The ultimate beginner's guide to firebase. Fireship.
- [Hakam et al. 2023] Hakam, Y., Ahessab, H., Gaga, A., and Hadadi, B. E. (2023). Renewable energy in smart grid: Photovoltaic power monitoring system based on machine learning using an open-source iot platform. *Lecture Notes in Networks and Systems*, 668 LNNS:241 – 251. Cited by: 0.
- [Santos 2022] Santos, S. (2022). Esp32 load cell (hx711) interface — random nerd tutorials. Random Nerd Tutorials.
- [Santos 2023] Santos, S. (2023). Esp32-cam save picture to microsd card and send to web server via wi-fi — random nerd tutorials. Random Nerd Tutorials.
- [Straub 2019] Straub, M. G. (2019). Balança arduino com célula de peso e hx711 - tutorial: Calibrando e verificando peso. UsinaInfo.
- [Unknown 2022] Unknown (2022). Esp32-cam camera module.
- [Yang et al. 2022] Yang, F., Zhang, J., Lei, Q., Li, G., Liu, J., and Wang, Z. (2022). Mudslide disaster monitoring and early warning system based on esp32. Cited by: 0.
- [Youness et al. 2022] Youness, H., Ahmed, G., and Haddadi, B. E. (2022). Machine learning-based smart irrigation monitoring system for agriculture applications using free and low-cost iot platform. page 189 – 192. Cited by: 0.