

VENDING MACHINE DE KITKAT UTILIZANDO SDK DO MERCADO PAGO A PARTIR DE AWS LAMBDA, COM ESP32

Isaac Fiuza Vieira¹, João Kaszuba², Thassiana Camilia Amorim Muller³, Vinicius Henrique Kielling⁴

Resumo

O projeto PixKat visa simplificar e agilizar a venda de chocolates KitKat utilizando tecnologia. O sistema é baseado em um microcontrolador ESP32 que se conecta a um display que permite a interação do usuário com o sistema. A integração de pagamentos é realizada através da plataforma Mercado Pago, utilizando a tecnologia Pix, gerenciada por um servidor AWS Lambda escrito em Python. A distribuição dos chocolates é automatizada por um servo motor que empurra o produto pela porta de liberação. Algumas peças e engrenagens do projeto são impressas em 3D, já o recipiente externo é uma caixa plástica simples. Para controle de versão do software, são utilizados Git e GitHub.

Abstract

The PixKat project aims to simplify and streamline the sale of KitKat chocolates using technology. The system is based on an ESP32 microcontroller that connects to a display to allow user interaction. Payment integration is carried out through the Mercado Pago platform, utilizing Pix technology, managed by an AWS Lambda server written in Python. The distribution of chocolates is automated by a servo motor that pushes the product through the release door. Some parts and gears of the project are 3D printed, while the external container is a simple plastic box. For software version control, Git and GitHub are used.

Palavras-chave: Vending Machine, API, Pix, Lambda-AWS, ESP32.

1. INTRODUÇÃO

Com a crescente utilização dos pagamentos instantâneos via Pix, o número de pequenos negócios descentralizados aumenta exponencialmente. Com isso, a fim de ampliar as regiões de vendas de pequenos varejistas, surge a oportunidade da criação de um modelo autônomo de máquina de vendas que integra a praticidade e eficiência da tecnologia de transações monetárias via Pix com a segurança de um recipiente fechado que protege o produto contra furtos. Neste projeto buscamos implementar esse modelo com o chocolate Kitkat, utilizando serviços bancários do Mercado Pago e servidores hospedados na *Amazon Web Services* (AWS).

2. ARQUITETURA DO SISTEMA

A arquitetura do projeto baseia-se em um modelo usuário, cliente (microcontrolador) e servidor (AWS). O fluxo de funcionamento entre o

usuário, cliente e servidor segue a seguinte sequência:

1. O usuário se aproxima da *vending machine*;
2. O usuário seleciona a quantidade de produtos desejados ao clicar nos botões “-” e “+” do display;
3. O usuário clica no botão “Confirmar” do display da *vending machine*;
4. O microcontrolador envia uma requisição POST com um JSON contendo a quantidade de chocolates selecionadas pelo usuário para o serviço de API *Gateway* da AWS;
5. A API invoca uma Lambda que, através de um SDK do Mercado Pago, gera um QR code com o valor a ser pago e um id para acompanhar o status de pagamento;
6. O microcontrolador recebe o id da transação e o QR code do pagamento a ser realizado;

¹ Acadêmico de Eng. de Computação. UTFPR, Pato Branco. E-mail: isaacvieira@alunos.utfpr.edu.br.

² Acadêmico de Eng. de Computação. UTFPR, Pato Branco. E-mail: joaokaszuba@alunos.utfpr.edu.br.

³ Acadêmica de Eng. de Computação. UTFPR, Pato Branco. E-mail: thassiana@alunos.utfpr.edu.br.

⁴ Acadêmico de Eng. de Computação. UTFPR, Pato Branco. E-mail: viniciuskielling@alunos.utfpr.edu.br.

7. O microcontrolador exibe o QR code do valor a ser pago no display;
8. O usuário escaneia o QR code com o aplicativo do Prestador de Serviços Bancários (PSP) de sua preferência e realiza o pagamento;
9. O microcontrolador envia uma requisição POST com um JSON contendo o id de pagamento para o serviço de API Gateway da AWS
10. A API invoca uma Lambda que, através de um SDK do Mercado Pago, confirma o pagamento realizado;
11. O microcontrolador recebe a confirmação do pagamento e aciona o servo motor que dispensa a quantidade de chocolates pagos.

A figura 1 apresenta um modelo esquemático do fluxo de funcionamento de interação entre as partes do projeto: usuário, cliente e servidor, que foi explicado anteriormente.

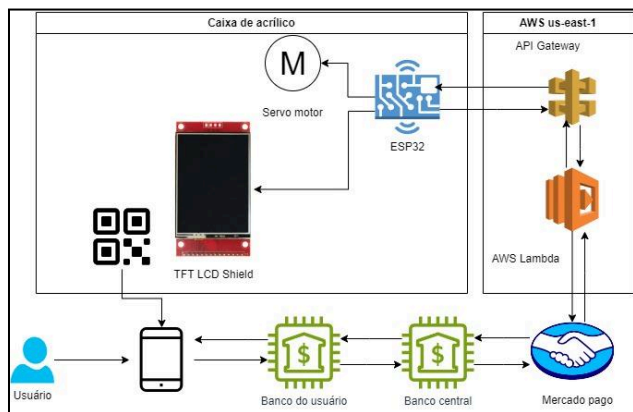


Figura 1: Fluxo geral do sistema.

3. MATERIAIS E MÉTODOS

Os materiais, métodos e tecnologias utilizados foram escolhidos baseados nos critérios de custo, simplicidade de operação e agilidade. Além disso, também foi levado em consideração a familiaridade que os projetistas tinham com as tecnologias empregadas e se os recursos destas seriam vantajosos para o desenvolvimento do projeto. Dessa forma serão detalhados a seguir os materiais utilizados para o desenvolvimento desse projeto e como estes foram empregados.

AWS LAMBDA

É um serviço de computação em nuvem no modo de operação *serverless*¹. Foi escolhido devido ao seu baixo custo e sua aplicação favorável às especificações deste projeto.

O serviço é responsável por realizar a comunicação entre o SDK do Mercado Pago e o API Gateway. A Lambda é composta por um código em Python que, quando recebe um JSON contendo o campo “value”, retorna um JSON com um campo “qrcode”, com a string para gerar o QR code e um campo “id”, com o número do id para verificar o pagamento. Além disso, quando a Lambda recebe um JSON com o campo “id” ela retorna um JSON com o campo “status” que será 1 para pagamento efetuado e 0 para pagamento não efetuado.

API GATEWAY

API Gateway é um serviço gerenciado pela AWS que permite a criação, monitoramento e publicação de serviços de API RESTful e WebSocket, no projeto, ele permite a comunicação entre o ESP32 e a AWS Lambda em tempo real.

No projeto, há apenas uma API com uma rota que recebe, a partir de um método POST, um JSON contendo a quantidade de produtos requisitados pelo cliente ou um identificador único de pagamento de uma transação Pix. O JSON é direcionado para a AWS LAMBDA que tratará os dados.

Todos os recursos de cloud foram hospedados nos servidores *us-east-1* no estado de Norte Virgínia - EUA.

ESP32 DEVKITV1

O ESP32 é um microcontrolador utilizado em diversos projetos de robótica e automação. Foi escolhido para fazer a integração de hardware e software do projeto pois, além de possibilitar a programação do controle da parte mecânica, também possui módulo Wi-Fi que permitia a comunicação com o servidor.

Para esse projeto, todo o código utilizado no microcontrolador ESP32 foi desenvolvido no ambiente de desenvolvimento do Arduino, a

¹ Modelo de execução de computação em nuvem no qual o provedor de nuvem aloca recursos de máquina sob demanda, cuidando dos servidores em nome de seus clientes. Wikipedia

ArduinoIDE em C/C++. Para tal, foram utilizadas algumas bibliotecas prontas para acelerar o desenvolvimento do projeto, dentre elas: ArduinoJson, WifiSecureClient, httpClient, ESP32Servo, Qrcode gen, MCUFRIEND, Adafruit TouchScreen e Adafruit GFX.

MOTOR TOWER PRO

O TowerPro MG995 360° é um servo motor de alto torque, com cerca de 15 Kg.cm e de rotação contínua. Sua função do projeto é ejetar o chocolate, após a confirmação do pagamento ser validado a partir da conexão entre o serviço Lambda AWS e o microcontrolador ESP32.

Quando a confirmação do pagamento é recebida, o motor começa a girar juntamente com a engrenagem circular que está acoplada em sua ponta. Essa engrenagem é chamada de pinhão e fica conectada a uma cremalheira, que transforma a rotação do servo em um movimento linear e assim consegue empurrar e puxar o pistão de liberação dos chocolates.

TFT LCD SHIELD 2.4"

O Display TFT 2,4" MCUFRIEND é um display *touch screen* resistivo com resolução de 240x320 e 18 bits de cores vivas. Sua utilização no projeto é primordial, visto que é responsável por toda a interação usuário/produto.

Seu uso mais importante no projeto é a exibição do QR code gerado para o pagamento dos chocolates. Porém o display também é utilizado para o usuário fazer a seleção de quantas unidades da mercadoria quer comprar e para exibir algumas telas informativas, como a tela de tempo restante de expiração do QR code, ou a tela de estoque insuficiente.

RECIPIENTE FÍSICO

Como base estrutural do projeto, foi utilizado um recipiente físico de polipropileno com dimensões de 33,00 cm x 23,00 cm x 18,00 cm. Essa caixa tinha como propósito guardar todos os componentes do projeto, tais como os produtos para venda, o microcontrolador, o sistema elétrico e o sistema mecânico. Isso pois, um dos propósitos fundamentais desse projeto era a segurança do sistema.

Outro ponto de interesse para esse projeto, era a versatilidade do produto para ser alocado em

diversos pontos de venda. Ao estruturar todo o projeto de maneira a ficar armazenado em uma pequena caixa, é possível garantir essa versatilidade. Visto que a caixa não ocupa muito espaço, ela pode ser alocada em qualquer lugar mesa em algum local onde ocorra a passagem de possíveis compradores.

MODELAGEM

Devido a necessidade de construir um sistema específico de armazenamento e liberação dos produtos comercializados, foi utilizado modelagem e impressão 3D para construir essas partes.

Para conter as unidades do chocolate, foi impresso um compartimento com capacidade para quinze chocolates, com dimensões de 6,6 x 9,4 x 15 centímetros. Também foi impresso uma base para esse compartimento, com o objetivo de elevar a plataforma dos chocolates na mesma altura que o pistão estava disposto. A *Figura 3* apresenta o modelo 3D da torre de chocolates e de sua base.

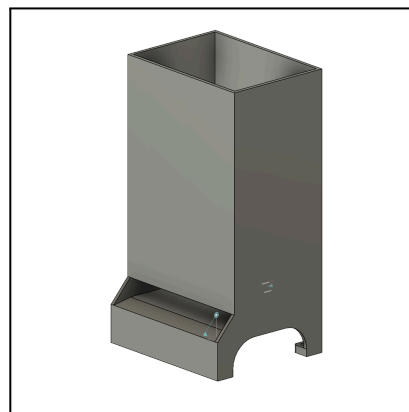


Figura 2: Modelagem da torre e sua base.

No sistema de liberação dos chocolates, foram modeladas e impressas quatro peças: a cabeça do pistão, a cremalheira, o pinhão e o *servo holder*. A cabeça do pistão tinha como função empurrar os chocolates quando a compra fosse aprovada. A cremalheira era quem empurrava a cabeça do pistão e ambos estavam ligados através de um parafuso. O pinhão estava ligado ao servo motor e era o responsável por mover a cremalheira para frente e para trás através da conversão da sua rotação em um movimento linear. Por fim, o *servo holder* era a peça responsável por segurar o servo motor e fixa-lo na estrutura física. A figura 4

apresenta o modelo 3D dessas quatro peças conectadas.

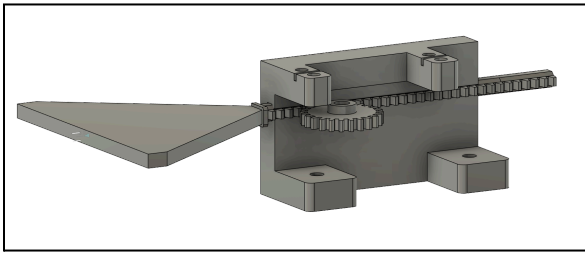


Figura 3: Modelagem do sistema de liberação.

A modelagem dos componentes foi desenvolvida através do software Fusion 360 da Autodesk. Já para a impressão 3D desses componentes foi utilizada a impressora Ender 3 Pro da Creality com o filamento PLA branco de 1,75mm de diâmetro. A figura 4 apresenta uma visão geral das peças impressas em 3D e de como elas foram posicionadas na caixa.

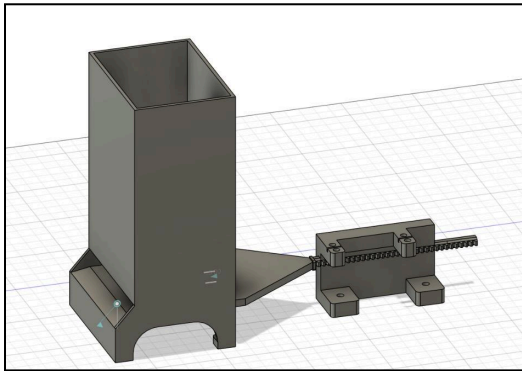


Figura 4: Modelagem dos componentes internos

4. RESULTADOS / MONTAGEM

O desenvolvimento da parte de software do projeto seguiu duas frentes principais. A primeira em relação a criação da API e sua hospedagem em um servidor da AWS e a segunda em relação a programação do microcontrolador ESP32 para controlar o display e o servo motor. Ainda houve uma terceira etapa, que se deu basicamente pela integração das duas primeiras etapas citadas.

Em relação à primeira etapa, essa se deu de forma muito satisfatória. Visto que um dos integrantes do projeto já havia trabalhado anteriormente com os servidores da AWS, essa etapa foi realizada de forma rápida e simples, sem maiores problemas.

A segunda etapa ocorreu de maneira similar a primeira, seguindo as especificações da arquitetura do sistema. O display foi configurado para exibir telas predefinidas dependendo de que

etapa da compra dos chocolates o usuário estava, o Wi-Fi foi configurado para se conectar a rede roteada pelo celular e o servo motor foi configurado para, ao ser confirmado o pagamento, ele executar a liberação dos chocolates. Porém, foi nessa etapa onde ocorreram alguns dos maiores desafios do projeto.

Como supracitado, foi escolhida a plataforma ArduinoIDE para fazer toda a programação do microcontrolador ESP32, isso pois, a plataforma disponibiliza uma vasta gama de bibliotecas prontas que serviram como acelerador do desenvolvimento de projetos. Mas ao fazer uso de diversas bibliotecas em um só código, alguns problemas de compatibilidade começaram a surgir, principalmente atrelados a as portas de entrada e saída do microcontrolador.

Dois outros problemas encontrados durante a programação do microcontrolador ESP32 foram: o excesso de uso de memória e a falta de paralelização do código. O alto consumo da memória do microcontrolador pode ser explicado por diversos fatores, porém é provável que o fator de maior impacto nesse projeto foi o uso de muitas variáveis globais. Já a falta de paralelização do código, principalmente na comunicação com o servidor, resultou em erros no tempo de exibição do QR code, devido a irregularidade do tempo de comunicação entre o servidor e o cliente que, ao depender da velocidade de conexão, pode se tornar instável.

A terceira etapa do desenvolvimento de software do projeto também foi um desafio, principalmente estabelecer uma comunicação segura e confiável entre o microcontrolador e o servidor. Foi necessário utilizar de um certificado de autenticação para liberar o acesso do microcontrolador ao servidor. O resultado disso foi o estabelecimento de uma comunicação eficaz entre o cliente e o servidor, com desempenho satisfatório e praticidade para o usuário final.

O desenvolvimento de hardware do projeto também ocorreu em duas etapas principais, uma relacionado ao sistema mecânico e outro relacionado ao sistema elétrico do projeto. Ambas essas etapas são intrinsecamente conectadas, visto que sem uma a outra não teria uso algum.

Como citado anteriormente, o projeto foi todo desenvolvido para ficar contido em uma caixa de pequenas dimensões. Além disso, também foi informado que algumas partes do projeto tiveram

que ser modeladas e impressas em 3D diante da necessidade de compartimentos e mecanismos de tamanhos limitados e funcionalidades específicas. Porém o desenvolvimento dessas peças em 3D não levou em consideração os aspectos físicos da caixa, tal qual as suas bordas arredondadas. Isso levou a que algumas funcionalidades não ocorressem de maneira esperada, como a liberação do chocolate que deveria sair por completo da caixa ao ser empurrado pelo pistão.

Mesmo com alguns problemas na montagem dos componentes impressos em 3D no recipiente físico, o sistema mecânico funcionava de maneira correta. Ao ser confirmada o pagamento dos chocolates o servo motor era acionado por um segundo em rotação anti-horária para empurrar o pistão e depois um segundo em rotação-horária para retrainr o pistão. Além disso, foi estabelecido um tempo de um quarto de segundo para que o comprador retirasse o chocolate antes que o servo fosse acionado mais uma vez, caso a compra fosse de mais de um produto por vez.

Para o protótipo, foi elaborada uma capa impressa em folha A4 para a tampa, com figuras e o preço de cada unidade do produto. A figura 5 apresenta uma foto dessa capa colada sobre a superfície superior da *vending machine*.



Figura 5: Vista do topo do recipiente

A etapa elétrica do hardware do projeto foi relativamente mais complicada que a etapa mecânica, principalmente pela presença do servo motor no sistema. Durante os testes iniciais do protótipo, o servo motor tinha sua alimentação oriunda do pino Vin do microcontrolador, da

mesma forma que o display LCD. Porém, toda vez que o servo finalizava uma ação, fosse de girar para empurrar ou retrainr o pistão, quando essa ação era finalizada, ocorria um pico de corrente no sistema que desligava a tela e, em alguns casos isolados, reiniciava o microcontrolador.

Diante dos problemas ao utilizar a mesma fonte de alimentação para servo motor, o microcontrolador e o display, foi optado por isolar a fonte de alimentação do servo motor. Para o microcontrolador e o display foi utilizado um *powerbank* com saída de cinco volts de tensão. Já para o servo motor, foi utilizado duas baterias de três vírgula oito volts de tensão conectadas em série. A figura 6 apresenta uma foto da vista lateral do protótipo, onde pode ser visto as conexões entre os componentes por meio dos *jumpers*.

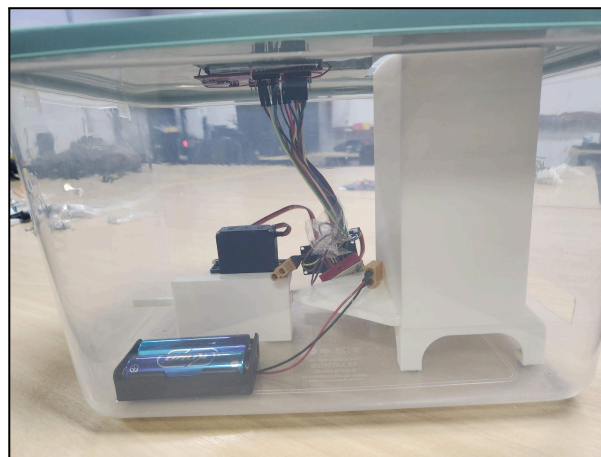


Figura 6: Vista lateral do recipiente

A união das etapas mecânicas e elétricas do projeto se deu de maneira relativamente simples, mas foi imprescindível para que o projeto funcionasse como o esperado. O resultado dessa integração foi melhor do que o esperado e todo o sistema funcionou de forma satisfatória e sem maiores problemas. A figura 7 apresenta uma foto da frente do protótipo, onde é possível ver a saída do chocolate, que desce pela torre de contenção do produto.



Figura 7: Frente do recipiente

Durante todo o desenvolvimento do projeto, o servidor manteve-se em funcionamento contínuo. Devido ao seu funcionamento *serverless*, os custos com uso dessa ferramenta foram praticamente nulos e a estimativa é que mesmo ao ampliar as operações realizadas, o custo continue relativamente baixo. Além disso, o microcontrolador e os outros componentes elétricos, como o display e o servo motor, também operaram de forma contínua, enquanto corretamente alimentados. Diante disso é possível dizer que os resultados obtidos pela implementação desse projeto foram satisfatórios e condizentes com o que era esperado ao início do seu desenvolvimento.

5. CONCLUSÃO

A partir dos resultados obtidos na construção deste projeto, conclui-se que foi possível realizar a integração de todas as tecnologias de *hardware* e *software* mencionadas para o desenvolvimento de um protótipo de *vending machine* para a venda do chocolate KitKat. Mas algumas questões, como a portabilidade, praticidade e otimização ainda podem ser debatidas e aprimoradas.

A respeito do recipiente utilizado, é notório que o recipiente do protótipo não é muito resistente e para o desenvolvimento de uma versão final e aprimorada, seria necessário fazer o uso de materiais mais resistentes e algumas travas de segurança. Também seria interessante aumentar a capacidade máxima de chocolates e o desenvolvimento de painel remoto de controle de estoque. Outro ponto que poderia ser aprimorado é a alimentação dos componentes de hardware através de uma tomada, que permitirá a operação ininterrupta do sistema.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Amazon API Gateway | Gerenciamento de APIs | Amazon Web Services. Disponível em: <<https://aws.amazon.com/pt/api-gateway/>>.

Amazon Lambda - AWS. Disponível em: <<https://aws.amazon.com/pt/lambda/>>.

Documentações Mercado Pago. Disponível em: <<https://www.mercadopago.com.br/developers/pt/docs#online-payment>>.

Fusion 360 for Personal Use | Fusion 360 | Autodesk. Disponível em: <<https://www.autodesk.com/products/fusion-360/personal>>.

Get Started - ESP32 - — ESP-IDF Programming Guide v5.2.1 documentation. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html>>.

ARDUINO IDE. Arduino Docs | Arduino Documentation. Disponível em: <<https://docs.arduino.cc/>>.

Display LCD TFT 2.4" Touchscreen Shield. Disponível em: <<https://www.autocorerobotica.com.br/display-lcd-tft-24-touchscreen-shield>>.

