



Escola Politécnica da Universidade de São Paulo
PSI 3542 - Sistemas Embarcados para IoT

Felipe Beserra (13683702)
Gabriel Kiyoki Hanashiro (13682020)

Projeto Final: Britney Cinnamon

São Paulo
2022

Felipe Beserra, Gabriel K. Hanashiro

Projeto Final: Britney Cinnamon

Este documento tem como objetivo apresentar o projeto elaborado por Felipe Beserra e Gabriel Hanashiro para o trabalho final da disciplina de IoT para Sistemas Embarcados.

Orientadores:
Prof. Sergio Takeo Kofuji
Fabrício Mansilla

São Paulo
2022

Resumo

Internet das Coisas (ou IoT) é a conexão dos objetos mundanos a internet, podendo ser aplicado em diversas áreas (médica, industrial, automação, etc). Aqui, descreve-se um projeto desenvolvido por alunos de graduação na Escola Politécnica da Universidade de São Paulo, apresentando um sistema completo de IoT cujas funcionalidades envolvem travas e lâmpadas inteligentes.

Sumário

Resumo	2
1. Requisitos do projeto final	4
1.1 Entregas	4
1.2 Requisitos	4
2. Projeto Final	4
2.1 Lista de Componentes	4
2.2 Descrição do sistema	5
2.3 Implementação do sistema	7
2.3.1 ESP client	7
2.3.2 RASP client	7
2.4 Utilização dos dispositivos	8
2.4.1 Modelos	8
2.4.2 Aplicativo e página web	10
3. Marketing	13
3.1 Comparação de mercado	13
3.2 Marca	14
4. Links importantes	14
4.1 Links relacionados ao grupo	14
4.2 Links de produtos para comparação de preços	15

1. Requisitos do projeto final

O trabalho final da disciplina de IoT para Sistemas Embarcados é projetar e aplicar uma solução completa de IoT usando dispositivos embarcados de IoT.

1.1 Entregas

O trabalho final têm como entregas:

- Minimum Viable Product (MVP);
- Folder do sistema;
- Apresentação do projeto (marketing e técnica);
- Vídeo tutorial no youtube do projeto, com detalhes do passo-a-passo da implementação do sistema. Deve conter uma demonstração de uso;
- Arquivos dos códigos publicados no GitHub;
- Tutoriais no GitHub.

1.2 Requisitos

O produto final deverá englobar:

- Um (ou mais) dispositivo embarcado de IoT para sensoriamento/atuação, com capacidade de comunicação MQTT;
- Um Smart IoT device/gateway/edge com capacidade de interação humano-máquina por voz para gerenciamento dos dispositivos de IoT embarcados e integração com serviços em nuvem;
- Uma aplicação de monitoramento em nuvem, capaz de conversar com usuários através de redes sociais e página web;
- Um aplicativo para smartphone android.

2. Projeto Final

2.1 Lista de Componentes

Os componentes usados no projeto foram:

- Raspberry Pi;
- ESP32;
- DHT11;
- Servo motor;
- Módulo relé 3.3V - 127V;
- LED (para o protótipo);
- Resistor (para o protótipo);
- Mini interruptor de alternância (para o protótipo);
- Borne 2 vias;
- Mini fonte 5V.

2.2 Descrição do sistema

O projeto final do grupo usa um ESP32 como dispositivo IoT que se comunica com o smart hub unicamente via MQTT. O dispositivo IoT controlado seria instalado dentro do interruptor do cômodo e controlaria a luz com um relé, de maneira integrada com o próprio interruptor (ou seja, a mudança de estado é o que altera a luz), um DHT11 para captar a temperatura e umidade, e um servo motor atuando como o mecanismo de uma trava.

A alimentação do dispositivo será realizada com os próprios cabos de alimentação da lâmpada na parede, apenas passando por um conversor de tensão para se adaptar ao ESP32 (para o caso do protótipo, teremos um carregador de celular para fazer essa conversão).

Por ser um protótipo e devido às diversas mudanças de design pelo qual o projeto passa, o grupo usou uma placa ilhada de fibra de vidro 10x5 centímetros para fazer o sistema de alimentação do ESP (mantendo espaço para os outros componentes, para ter uma ideia de tamanho) e usou uma protoboard e um LED para realizar os testes e comprovar o funcionamento, imagens 1 e 2 respectivamente. Além disso, por não termos um microfone compatível com o Raspberry Pi, rodamos o código do smart hub no computador, porém ele é compatível com o mini-computador (apenas alterando o contexto do rhino para ser compatível com o sistema).

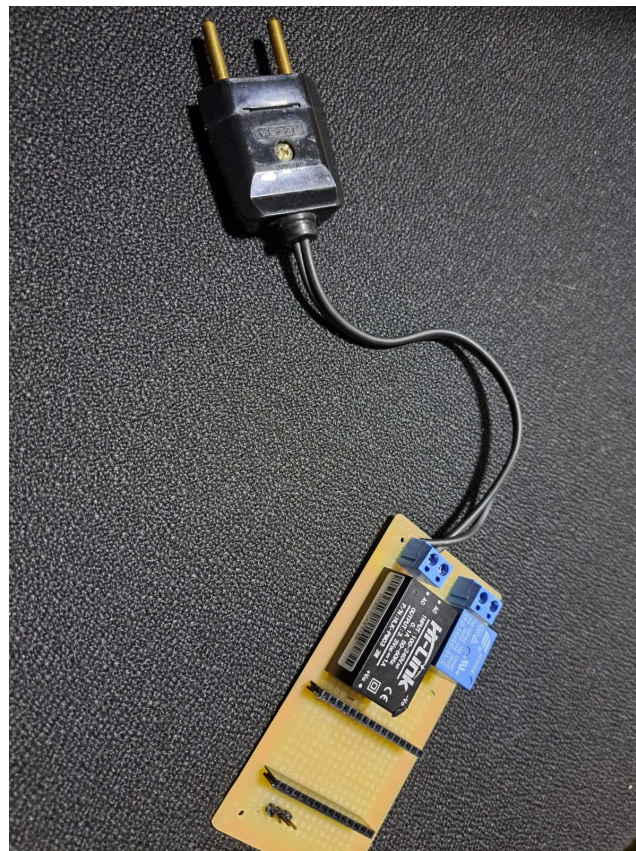


Imagem 1 - Protótipo do dispositivo IoT na protoboard.

FONTE: Autoral.

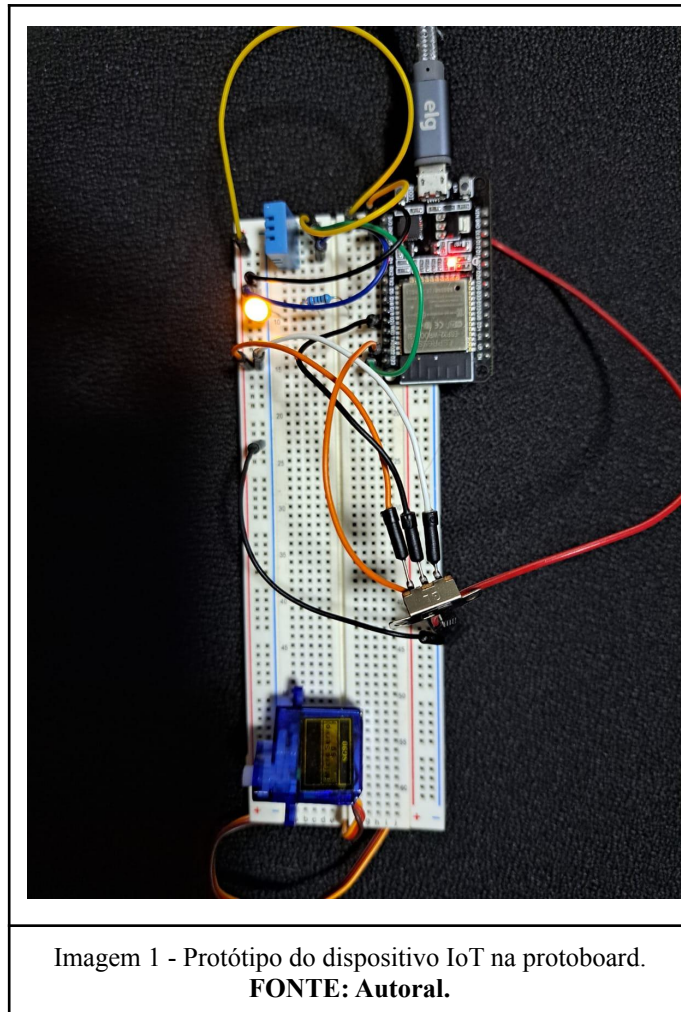


Imagem 1 - Protótipo do dispositivo IoT na protoboard.
FONTE: Autoral.

A comunicação MQTT é realizada utilizando o broker EMQX, um serviço open source. A estrutura de comunicação, incluindo o broker, pode ser vista na imagem 2.

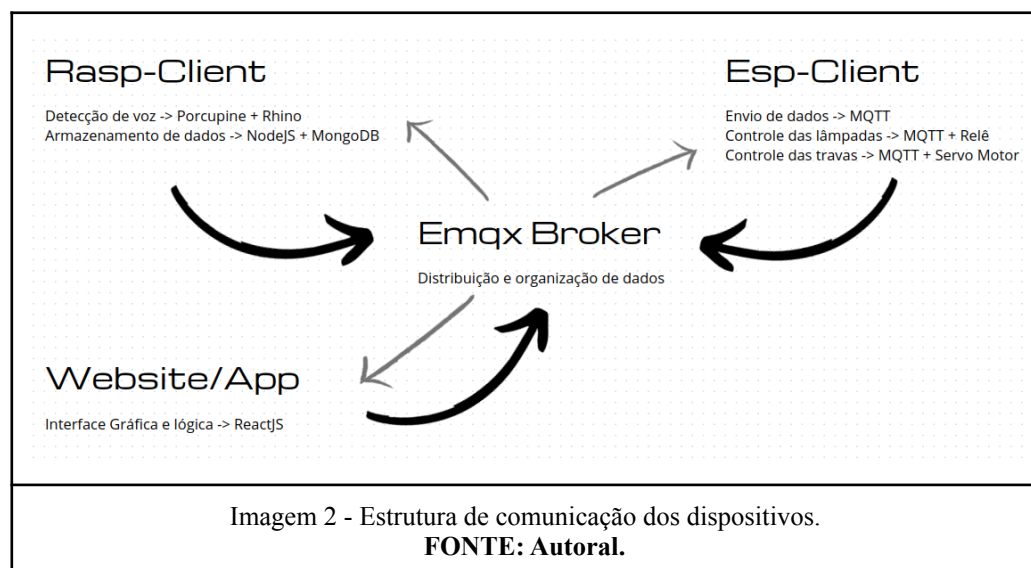


Imagem 2 - Estrutura de comunicação dos dispositivos.
FONTE: Autoral.

O smart hub, no caso o raspberry pi, conecta na internet e se comunica com o ESP32 via MQTT, como mencionado anteriormente. Ele é a interface para comunicação humano-máquina, ou seja, interação por voz para controle do dispositivo IoT, funcionando com o PicoVoice (usando Porcupine e Rhino).

Temos apenas a palavra “Cinnamon” como wake word do dispositivo, pois “Britney” não está incluído no dicionário do PicoVoice. Apesar do porcupine funcionar com uma palavra em inglês, o speech-to-intent do Rhino funciona de acordo com a língua portuguesa; treinamos o contexto dele para obedecer aos comandos de ligar, desligar, trancar e destrancar.

2.3 Implementação do sistema

2.3.1 ESP client

O ESP32, como mencionado anteriormente, envia dados de temperatura e umidade do DHT11, o estado da lâmpada e da trava a cada 5 segundos, aproximadamente, além de um aviso toda vez que liga; e recebe comandos de mudança de estado da lâmpada e da trava. Ambos trocando informações com o Raspberry Pi via MQTT (todos os dados são enviados no formato JSON, sigla para *JavaScript Object Notation*).

Para isso, existem os códigos boot.py (para conexão com internet) e o código main.py, que utilizam as bibliotecas components.py (escrita pelo grupo especificamente para este projeto) e mqttsimple.py (de uso público). No main.py, há, inicialmente, uma rotina para tentar uma conexão com o broker MQTT, que caso falhe, o dispositivo reinicia e tenta de novo. Em seguida, são definidas funções para verificar se há novas informações em algum tópico MQTT e para mandar os dados do DHT11 que serão armazenados no Raspberry Pi para possíveis análises dos dados. Além delas, uma última função chama a rotina dos componentes para realizar a coleta de dados do DHT e o funcionamento dos outros componentes de acordo com as mudanças, ou falta delas.

No caso do interruptor inteligente, diferentemente de um comum, não há um estado ligado e um estado desligado, mas sim, um funcionamento semelhante a de um push button, pois a mudança de estado (apertando a switch, independente de direção) acende ou apaga a luz.

2.3.2 RASP client

O Raspberry Pi é o smart hub, isto é, o centro de inteligência que atua como assistente controlado por voz, host do servidor do app e dos outros serviços necessários para o funcionamento do dispositivo. Para tal, ele possui uma rotina para iniciar a comunicação MQTT, e uma conexão direta com o MongoDB (software de banco de dados).

Além disso, tem-se uma implementação do sistema porcupine, que serve para identificar a wake word (treinado com a palavra “Cinnamon”) e rhino, um software de

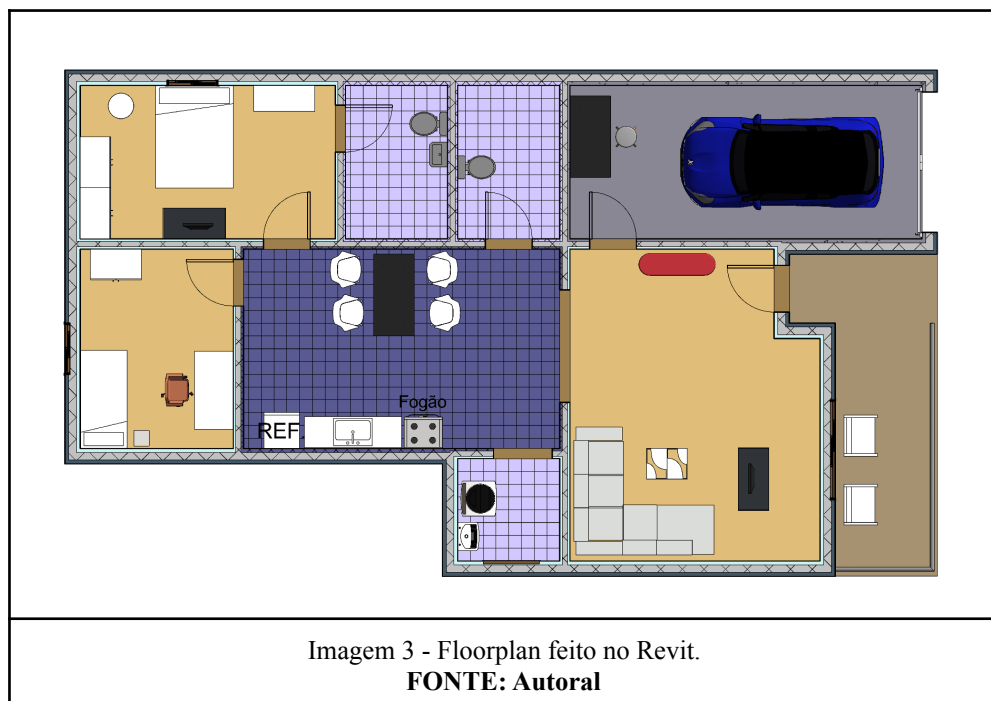
speech-to-intent (com o contexto elaborado pelo grupo) para identificação de voz, ambos serviços gratuitos, porém limitados, do PicoVoice.

Por fim, o Raspberry Pi também possui a rotina necessária para rodar a página da web e o aplicativo.

2.4 Utilização dos dispositivos

2.4.1 Modelos

Para utilizar o sistema, primeiramente, elaboramos uma casa fictícia usando o aplicativo Revit da Autodesk para modelar o Floorplan da imagem 3, ele possui 2 quartos, 1 cozinha, 2 banheiros, 1 garagem, 1 sala e 1 área de serviço.



Para visualização, o grupo fez um modelo 3D usando o SolidWorks de acordo com o quarto de um dos integrantes do grupo, como pode ser visto na imagem 4 abaixo. Vemos que apenas o motor e o sistema da trava no geral devem ficar para fora da parede, que seria conectada via cabos com o dispositivo dentro do interruptor.

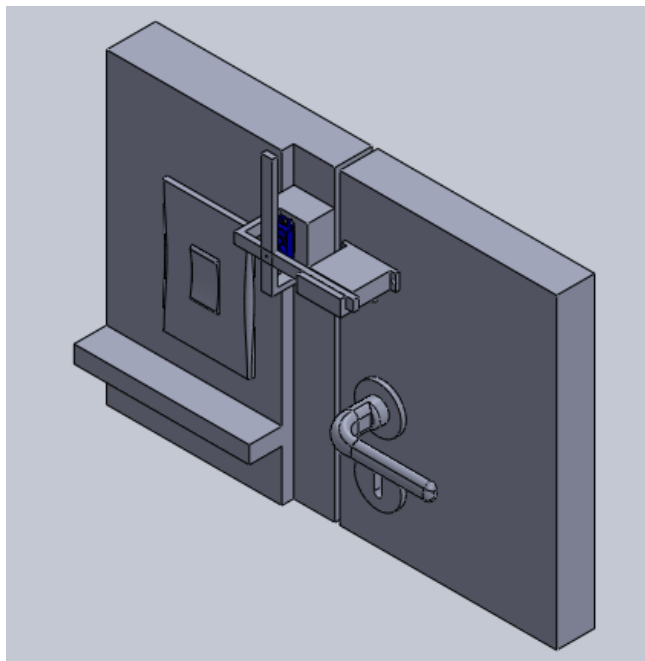


Imagem 4 - CAD da montagem da trava na parede

FONTE: Autoral

Vale mencionar que na imagem acima, os dois estados da trava (trancado e destrancado) estão sendo exibidos ao mesmo tempo, quando na realidade, teria apenas uma barra servindo como tranca. O modelo, para evitar ter toda a força exercida sobre o servo motor, possui uma extensão da barra em um formato de gancho para se conectar com o próprio suporte do motor, como se pode ver na imagem 5 abaixo.

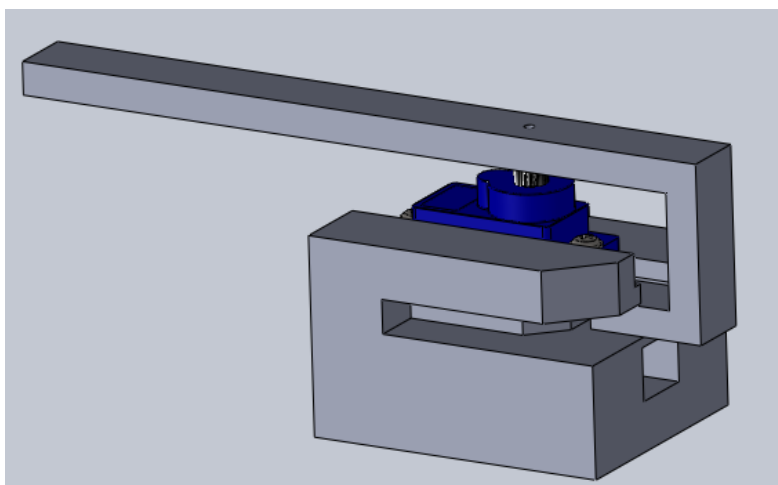


Imagem 5 - CAD da montagem do servo motor.

FONTE: Autoral

2.4.2 Aplicativo e página web

A página da web foi feita usando a ferramenta “React”, de modo que ela consiga servir como um aplicativo de celular também, criando um atalho nele que leva ao site.

Tanto o aplicativo quanto a página da web seguem um mesmo princípio, o floorplan é exibido em uma tela com botões sobre cada cômodo controlando a luz (no caso da cozinha, sala, garagem e os quartos) e 2 botões controlando as travas (no caso dos dois quartos). No caso do protótipo, como há apenas um cômodo funcionando, há apenas 2 botões na tela (uma lâmpada e uma trava).

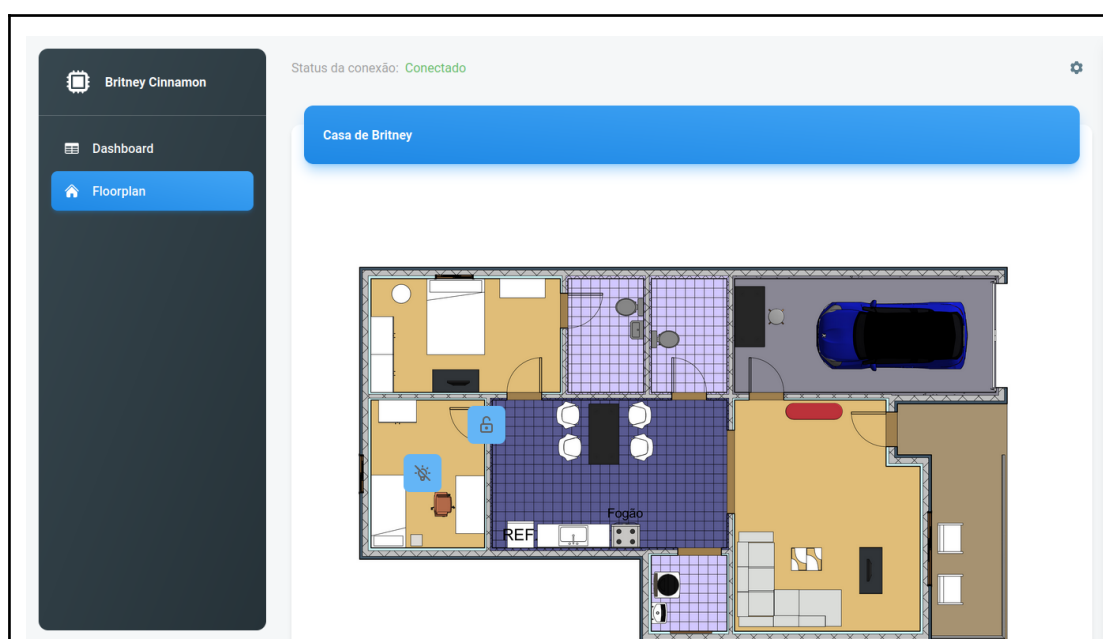


Imagem 6 - Aba da página web com o floorplan (lâmpada apagada).

FONTE: Autoral

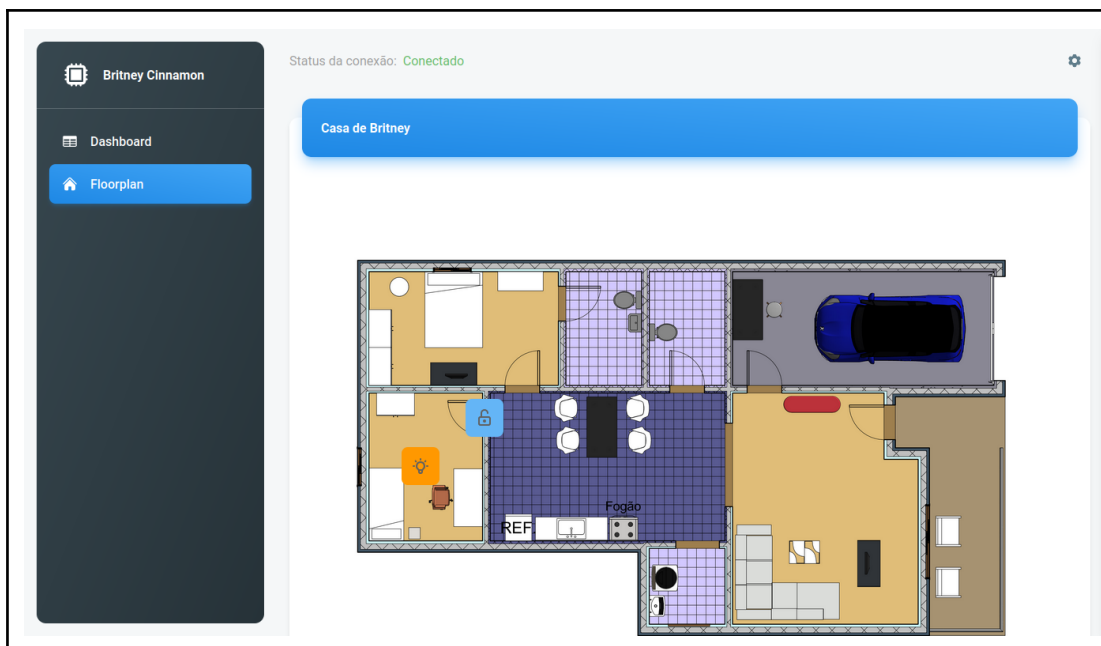


Imagem 6 - Aba da página web com o floorplan (lâmpada acesa).

FONTE: Autoral

Em uma aba separada (tanto no celular quanto na página da web), temos o dashboard, no qual aparece a média dos últimos 10 dados coletados da temperatura e umidade recentes, e logo abaixo deles, todos os valores sendo exibidos em formato de gráficos. Além disso, temos uma área exibindo a quantidade de dispositivos (clients) online e um gerador de número aleatório que muda toda vez que o site recebe um pacote (atua como um indicador da atualização dos dados).

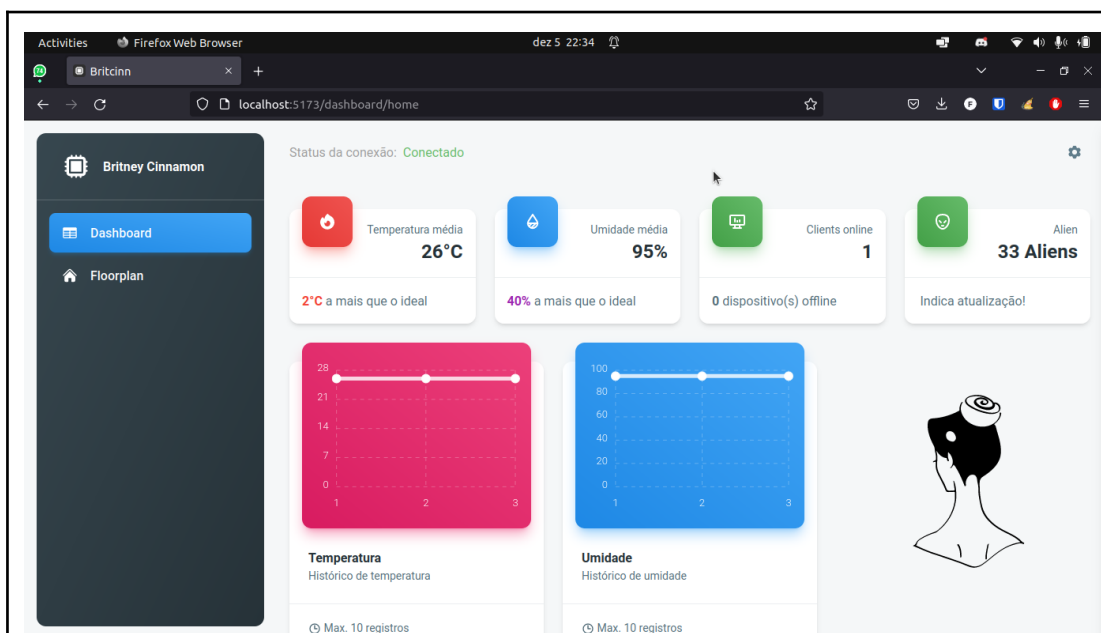


Imagem 7 - Aba da página web dashboard com o link

FONTE: Autoral

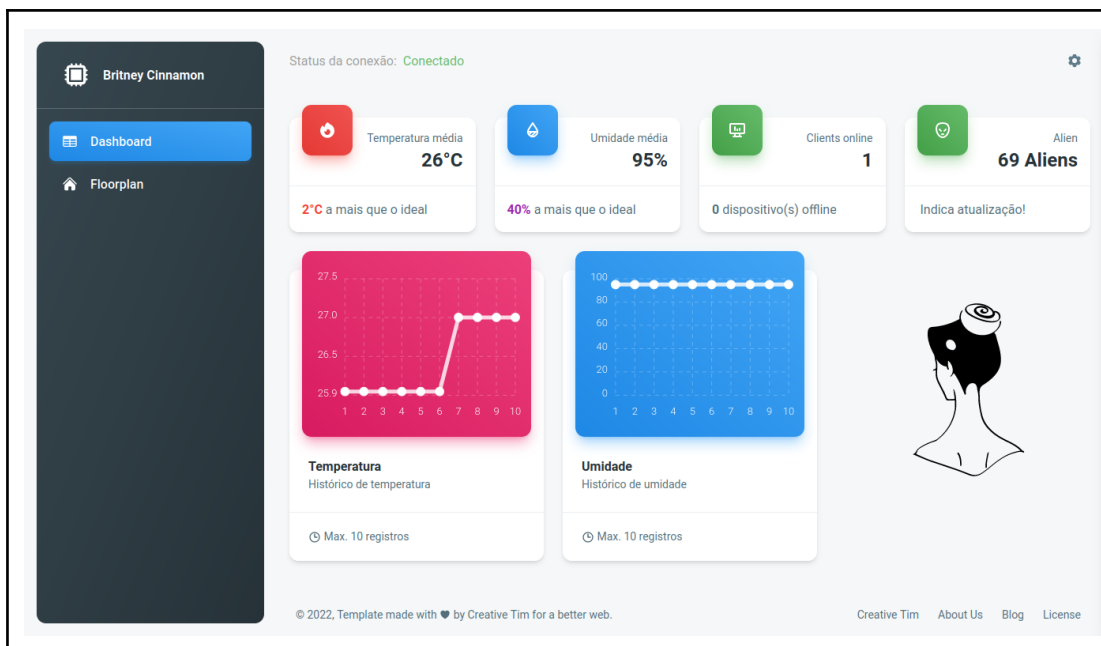


Imagem 7 - Aba da página web dashboard em tela cheia.
FONTE: Autoral

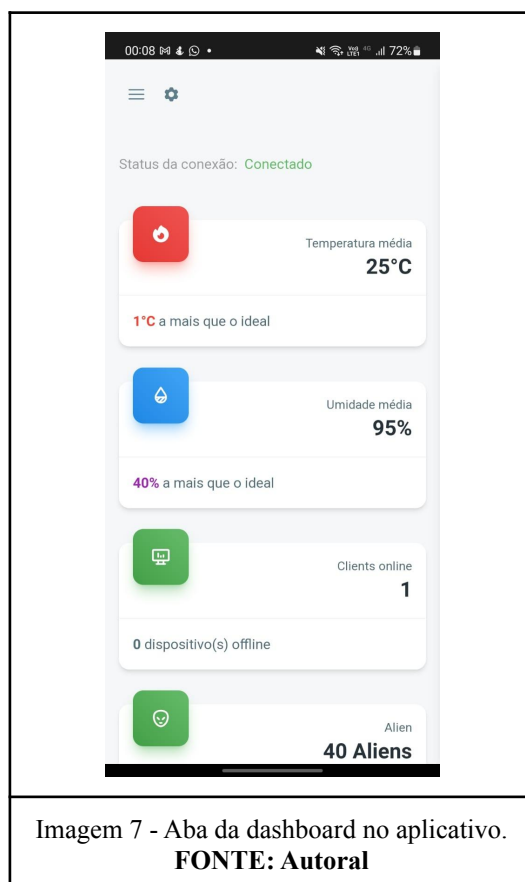


Imagem 7 - Aba da dashboard no aplicativo.
FONTE: Autoral

3. Marketing

3.1 Comparação de mercado

O projeto final visa recriar os assistentes controlados por voz como a Alexa da Amazon e o Google Home Mini da Google e os dispositivos que eles são capazes de manipular. O resultado apresentado possui um grande valor acadêmico e um potencial econômico também. Feito por estudantes de graduação universitários, usando majoritariamente softwares e outros serviços open-source e gratuitos (todos são open-source, com exceção do PicoVoice), além de open-hardware também. Isto demonstra uma expansão na distribuição e oferta de tecnologia, como também a existência de aplicativos gratuitos que possuem recursos de qualidade aplicáveis em projetos pessoais capazes de simular, até certo ponto, os produtos de grandes empresas, como a Google e a Amazon.

Economicamente falando, atualmente o projeto é incapaz de competir com grandes empresas, no total, o projeto final teve um custo de R\$792,00 (R\$680,00 do raspberry pi 3; R\$40,00 do ESP32; R\$20,00 do servo motor; R\$14,00 do DHT11; R\$10,00 do relé; e R\$8,00 da placa ilhada), sendo que o maior gasto foi o Raspberry pi, pois está passando por uma alta de preços devido a falta global de silício; enquanto um google home mini pode ser encontrado por aproximadamente R\$200,00 e um echo dot da 4ª Geração por R\$400,00, estes com mais funções e sensores que o nosso projeto.

A vantagem, porém, deriva não do preço mas da capacidade de manipulação pelo usuário que o projeto desenvolvido possui. Enquanto os produtos comerciais visam uma privacidade do sistema que desenvolvem, o que acaba limitando o quanto se pode alterar as funcionalidade do dispositivo; o projeto do grupo apresenta uma capacidade de manipulação pelo usuário muito maior, um atrativo para muitos makers que desejam implementar um assistente controlado por voz em seus projetos pessoais.

Outro ponto que se torna um atrativo no projeto é o dispositivo IoT. Este custou um total de R\$92,00, enquanto outros em circulação, para somar uma aplicação semelhante a nossa, custaria em torno de R\$340,00 (com uma lâmpada inteligente custando R\$40,00; um interruptor inteligente R\$100,00; e uma trava inteligente R\$200,00). Disso, percebemos que há um aspecto economicamente competitivo desenvolvido, o dispositivo IoT, cujo preço é muito menor, também tendo um aspecto customizável pelo cliente, apesar de certas limitações de processamento.

Analisando em mais detalhes essa conclusão que obtivemos, temos que usando diversos dispositivos COTS (commercial-off-the-shelf), ou seja, dispositivos prontos, alcançamos um produto não otimizado, ou seja, um produto funcional, porém com desperdícios ao longo do caminho, uma vez que nenhum produto foi feito especificamente para atuar neste projeto em específico. Porém, mesmo com esses fatores, criou-se uma alternativa consideravelmente mais barata para dispositivos inteligentes capazes de comunicação e controle via internet, além de apresentar um fator de manipulação pelo usuário das funcionalidades dele.

Um aspecto interessante quanto ao nosso projeto é que o dispositivo IoT engloba 3 produtos comerciais em 1 (trava, lâmpada e interruptor inteligente), em um único controlador, algo não muito comum devido às limitações físicas e de processamento do dispositivo, mas que se bem aplicado pode trazer uma barateamento do produto sem queda de performance.

Em conclusão, percebemos que o produto tem grande valor acadêmico ao simular produtos comerciais de grandes empresas com recursos limitados e utilizando softwares e outros serviços gratuitos. Além disso, temos uma possibilidade de transformá-lo num produto economicamente viável, reapropriando elementos de funcionalidades mais amplas (como o ESP32 e o Raspberry pi) e manipulando diversos dispositivos com eles, mantendo ainda um aspecto mutável deles, ou seja, capacidade de alteração de código pelo usuário, e gerando um elemento funcional mais barato que os anteriores.

3.2 Marca

Para se tornar um produto comercial e competitivo, é necessário uma produção de componentes mais direcionados para a nossa aplicação, ao invés de usar microcontroladores gerais. Além disso, é necessário um foco em desenvolver produtos modulares e programáveis, com um diferencial em relação aos competidores, atraindo um nicho da população que gosta de manipular seus próprios dispositivos.

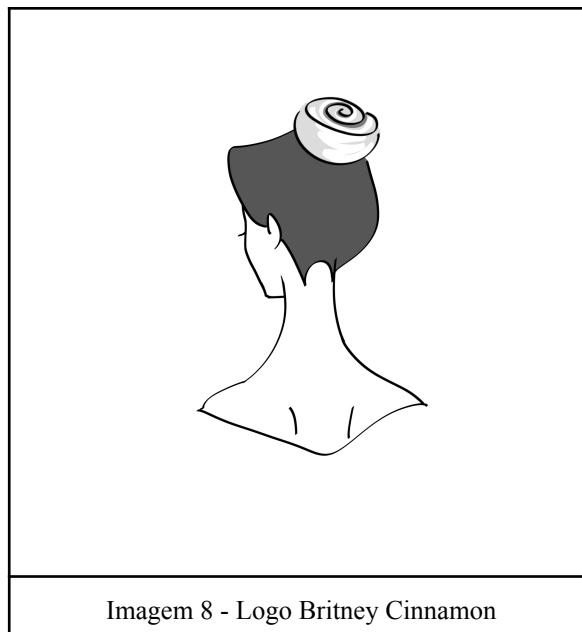


Imagem 8 - Logo Britney Cinnamon

4. Links importantes

4.1 Links relacionados ao grupo

Repositório do GitHub principal: <https://github.com/Beserrovsky/britcinn>

<https://github.com/Beserrovsky/britcinn-website>

https://drive.google.com/drive/u/0/folders/1Ou_RRr1SyYoGhryZdT9D951gfnK7g_yX

Vídeo Apresentação: <https://youtu.be/LhZvgNEXjJY>

Vídeo Passo-a-passo da implementação do projeto: https://youtu.be/GOQ3h5-p_FI

E-mail dos participantes: felipebeserra25@usp.br; gabriel.hanashiro@usp.br.

4.2 Links de produtos para comparação de preços

https://produto.mercadolivre.com.br/MLB-2755111466-tranca-inteligente-xiaomi-blutooth-para-portas-e-gavetas-_JM?matt_tool=86477614&matt_word=&matt_source=google&matt_campaign_id=14302215516&matt_ad_group_id=134553699508&matt_match_type=&matt_network=g&matt_device=c&matt_creative=539425477807&matt_keyword=&matt_ad_position=&matt_ad_type=pla&matt_merchant_id=165339711&matt_product_id=MLB2755111466&matt_product_partition_id=1402952723409&matt_target_id=pla-1402952723409&gclid=Cj0KCCQiAvqGcBhCJARIsAFQ5ke6_BDJOCX-Ub-txZAVmqphdAldDIWKToDMYzPptRRhRYta6VvT3E6q4aAhZREALw_wcB

https://curtocircuito.com.br/placa-doit-esp32-esp32-wroom-32d-wifi-bluetooth.html?utm_term=&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=7016354091&hsa_cam=16065462870&hsa_grp=130180187942&hsa_ad=579259865430&hsa_src=g&hsa_tgt=pla-490064249712&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQiAvqGcBhCJARIsAFQ5ke5tHuk-rXEky6tsaqegT0lR7gvq3RRFcoNwQ0yJAeuU3I5zsGMQ1EAaAip6EALw_wcB

https://www.amazon.com.br/Placa-Raspberry-Quadcore-1-2ghz-Bluetooth/dp/B01CD5VC92/ref=asc_df_B01CD5VC92/?tag=googleshopp00-20&linkCode=df0&hvadid=379816377800&hvpos=&hvnetw=g&hvrnd=7686807583425085841&hvpone=&hvptwo=&hvmqt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1001773&hvtargid=pla-344941896120&psc=1

<https://www.baudaeletronica.com.br/placa-padr-o-tipo-ilha-10x5-cm.html>

<https://www.baudaeletronica.com.br/sensor-de-umidade-e-temperatura-dht11.html>
https://www.americanas.com.br/produto/2388979154?epar=bp_pl_00_go_infaces_d_60_n_comp_oi5&opn=YSMESP&WT.srch=1&offerId=5f8f7b421b186381b5dc0991

[&gclsrc=aw.ds&gclid=Cj0KCQiAvqGcBhCJARIsAFQ5ke6-nFliadjj88Fy7oeqK-KR8UgHOPR0816nYJraDkxFozEgepijrpYaAuGjEALw_wcB](#)

[https://www.amazon.com.br/Novo-Echo-Dot-4%C2%AA-gera%C3%A7%C3%A3o/dp/B084DWCZY6/ref=asc_df_B084DWCZY6/?tag=googleshopp00-20&linkCode=df0&hvadid=404840237192&hvpos=&hvnetw=g&hvrnd=3284713935391369917&hvpon=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1001773&hvtargid=pla-989629063328&psc=1](#)

[https://www.baudaeletronica.com.br/modulo-rele-5v.html?gclid=CjwKCAiAhKycBhAQEiwAgf19er6s_PDxv_VfKxnNNhZ9nfbtBYiZo-4oIgel2Z9VZnqrxR-8SrKQAxoCJ04QAvD_BwE](#)