

PROTÓTIPO DE AUTOMAÇÃO RESIDENCIAL UTILIZANDO UMA ASSISTENTE DE VOZ

Aluno(a): Leandro Dallarosa Neto

Orientador: Miguel Alexandre Wisintainer

Roteiro

- Introdução;
- Objetivos;
- Fundamentação Teórica;
- Trabalhos Correlatos;
- Requisitos;
- Especificação;
- Implementação;
- Operacionalidade e Implementação;
- Resultados e Discussões;
- Conclusões e Sugestões.

Introdução

- Assistentes inteligentes de voz em dispositivos móveis;
- Automação residencial:
 - Comodidade;
 - Segurança.



Objetivos

Controlar periféricos de uma casa automatizada por meio de comandos de VOZ:

- controlar o estado de uma luz;
- consultar a temperatura do ambiente;
- abrir uma porta eletrônica.

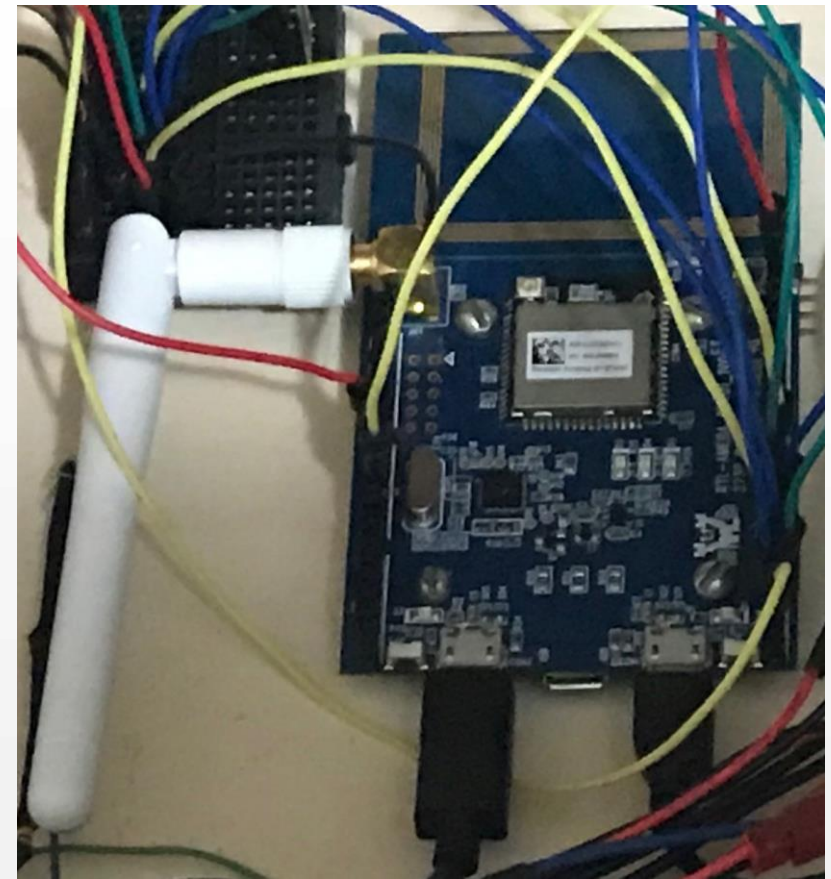
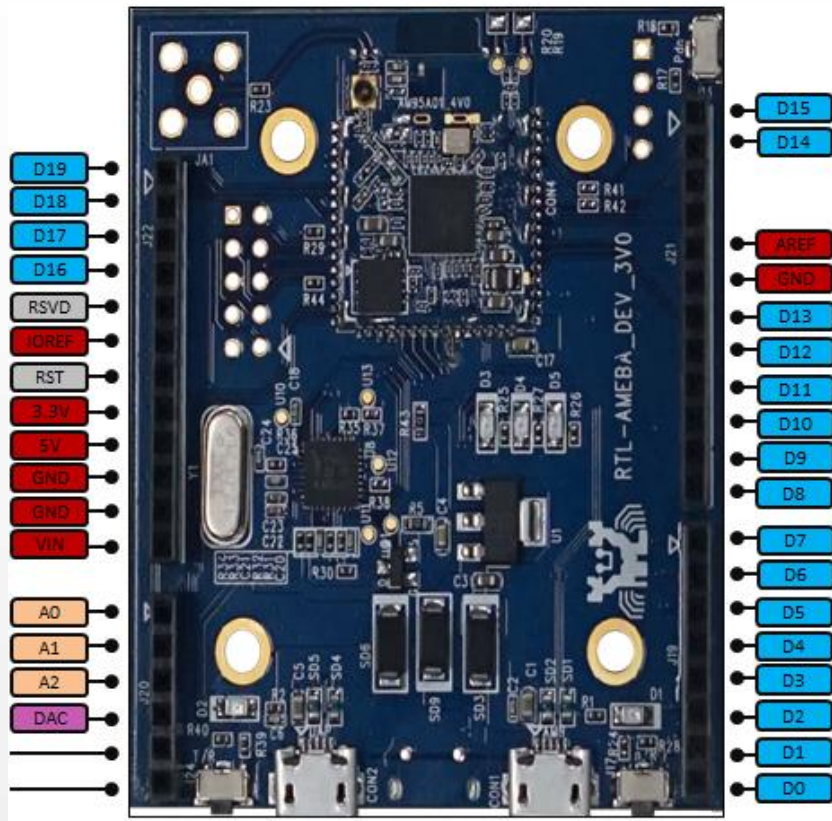
Fundamentação Teórica

- Arduino Ameba
 - CPU Arm;
 - WiFi;
 - Tag NFC;
 - Segurança em hardware.



Fundamentação Teórica (cont.)

- Arduino Ambeba



Fundamentação Teórica (cont.)

- Amazon Web Services

AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage).

Recently visited services

- IoT Core
- CloudWatch
- Lambda
- Certificate Manager
- IAM

All services

- Compute**
 - EC2
 - Lightsail
 - Elastic Container Service
 - EKS
 - Lambda
- Management Tools**
 - CloudWatch
 - AWS Auto Scaling
 - CloudFormation
 - CloudTrail
 - Config
- Mobile Services**
 - Mobile Hub
 - AWS AppSync
 - Device Farm
 - Mobile Analytics

Fundamentação Teórica (cont.)

- Amazon Alexa



amazon alexa



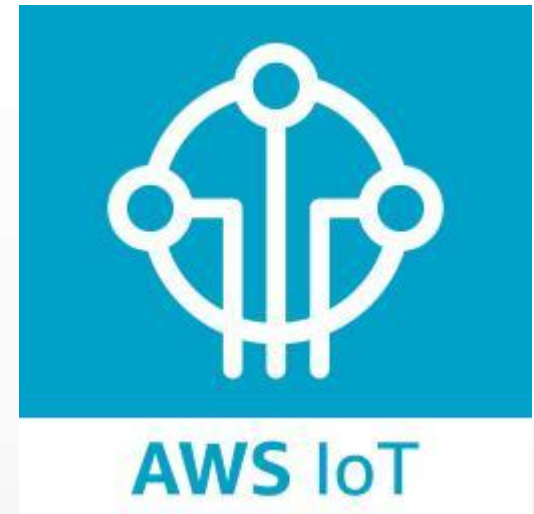
Fundamentação Teórica (cont.)

- AWS Lambda
 - Pagar pelo que usar;
 - Disponibilidade;
 - Segurança.



Fundamentação Teórica (cont.)

- AWS IoT
 - Shadow (sombra);
 - Segurança.



Trabalhos Correlatos

TRABALHO/ CARACTERÍSTICA	MULTIPLATAFORMA	NECESSIDADE DE INTERNET	ACIONAMENTO POR VOZ
Botke (2014)	sim	sim	não
Beghini (2013)	não	sim	não
Lisboa et al (2014)	sim	sim	não

Requisitos Funcionais

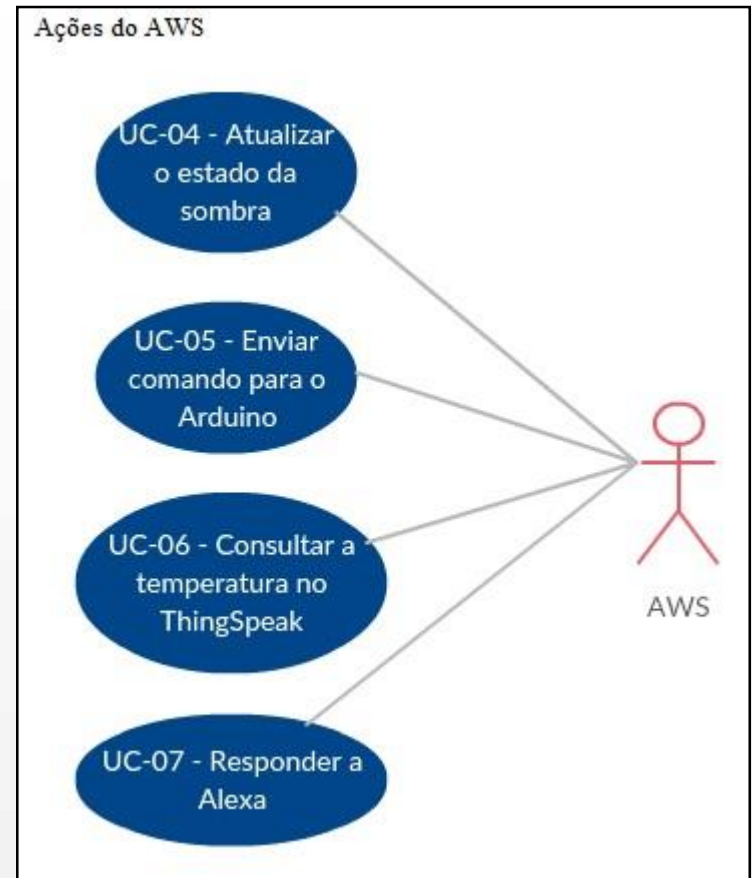
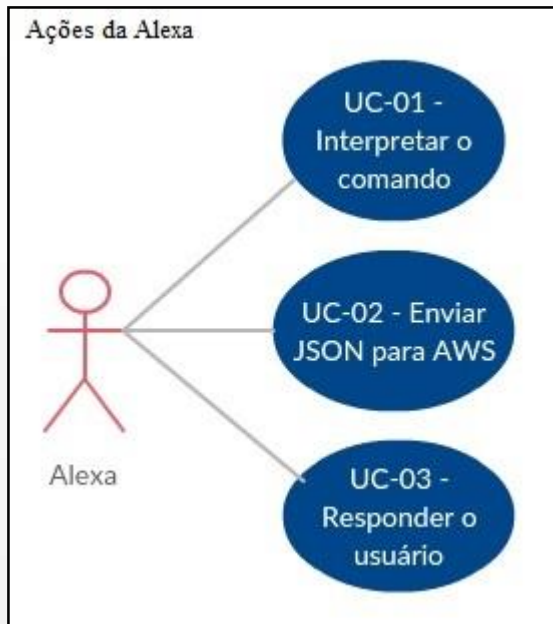
- A aplicação deverá permitir que o usuário acenda e apague uma luz da residência por comando de voz;
- a aplicação deverá permitir consultar a temperatura do ambiente por comando de voz;
- a aplicação deverá permitir abrir uma porta eletrônica por comando de voz.

Requisitos Não Funcionais

- A aplicação deverá ser desenvolvida na plataforma do Arduino Ameba;
- a aplicação deverá utilizar a plataforma da Amazon Web Services (AWS).

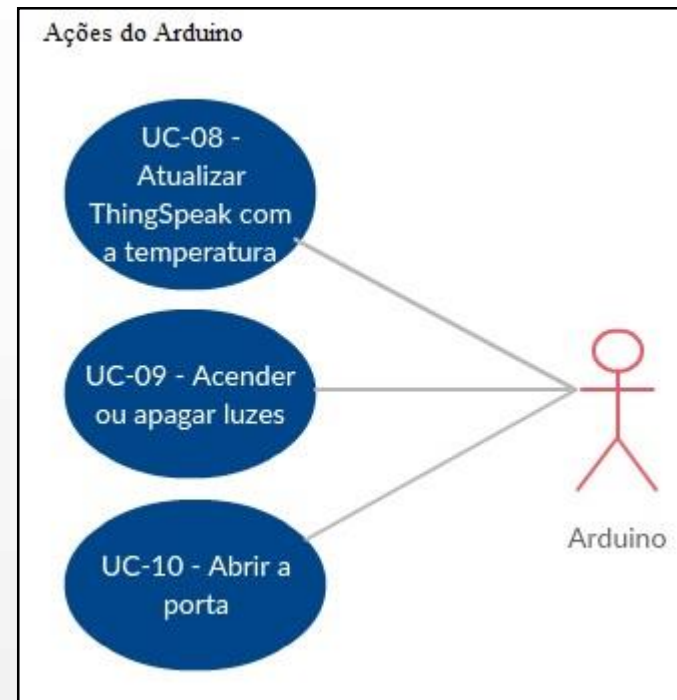
Especificação

- Diagrama de Casos de Uso

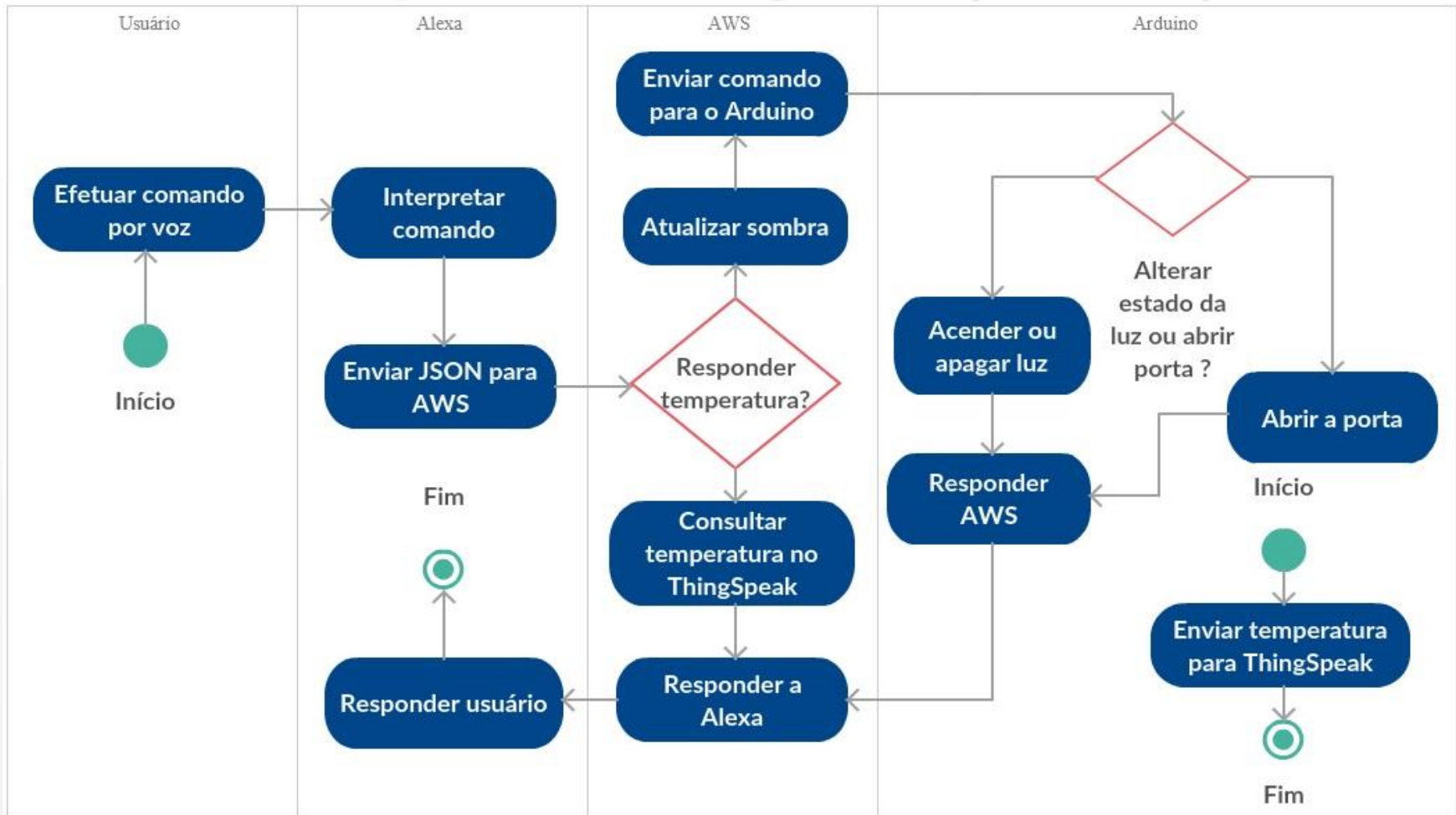


Especificação (cont.)

- Diagrama de Casos de Uso

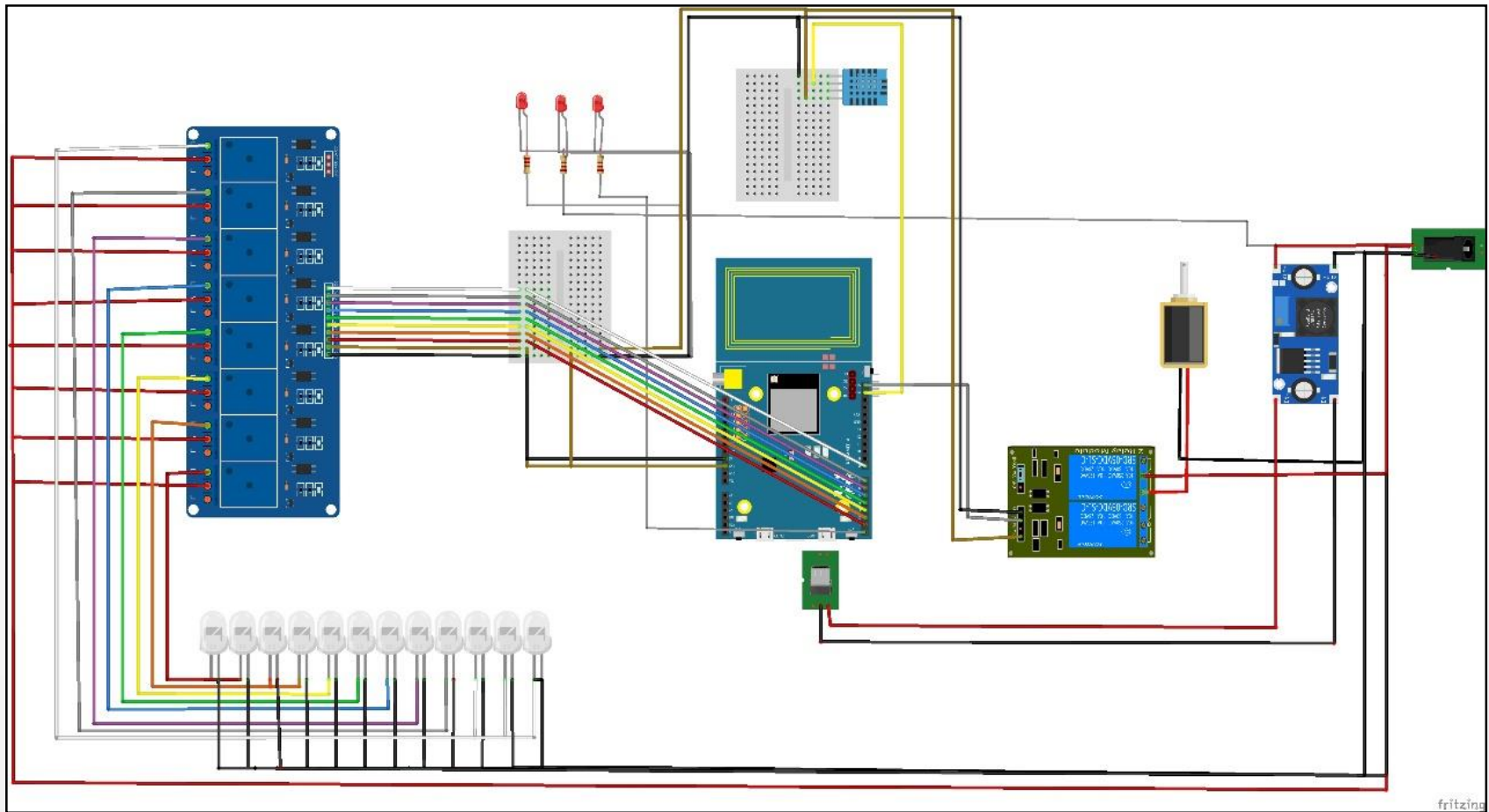


Especificação (cont.)



- Diagrama de Atividades

Especificação (cont.)



- Diagrama esquemático

Implementação

- Arquitetura do protótipo



Implementação

- Função Lambda



ControlLight



Kit Alexa Skills



Adicione os gatilhos a partir da lista à esquerda



AWS IoT



Amazon CloudWatch Logs

Os recursos aos quais a função tem acesso serão mostrados aqui

Implementação

AWS Lambda

Código da função [Informações](#)

Tipo de entrada de código

Editar código em linha

Runtime

Node.js 6.10

Mani

ind

```
File Edit Find View Goto Tools Window
Environment
  ▾ ControllLight
    index.js
index.js
205     payload : { state : { desired : { status :0, output : + cardTitle + }}}
206   };
207
208   //Update Device Shadow
209   iotData.updateThingShadow(paramsUpdate, function(err, data) {
210     if (err){
211       console.log(err, err.stack);
212
213       speechOutput = "fail to update thing shadow";
214       repromptText = "fail to update thing shadow";
215       callback(sessionAttributes,buildSpeechletResponse(cardTitle, speechOutput, repromptText, shouldEndSession));
216     }
217     else {
218       console.log(data);
219
220       sessionAttributes = createStateAttributes(0);
221       speechOutput = "Ok, it's done";
222       repromptText = "Done";
223       callback(sessionAttributes,buildSpeechletResponse(cardTitle, speechOutput, repromptText, shouldEndSession));
224     }
  }
```

Implementação



Interaction Model

Invocation

Intents (11) + Add

- IntentOne 🗑️
- PinState 🗑️
- IntentTwo 🗑️
- IntentThree 🗑️
- IntentAll 🗑️
- IntentFour 🗑️
- IntentFive 🗑️
- IntentSix 🗑️
- IntentSeven 🗑️
- IntentEight 🗑️
- IntentFifteen 🗑️
- Temperature** 🗑️

Built-In Intents (0)

Slot Types (2) + Add

- LIGHT_STATE 🗑️
- PIN_STATE 🗑️

JSON Editor

Intents / Temperature

Sample Utterances (4) ?

What might a user say to invoke this intent?

- to get the temperature
- to get temperature
- get the temperature
- get temperature

Intent Slots (0) ?

ORDER ?	NAME ?
1	Create a new slot

Intent Confirmation

Does this intent require confirmation? ?

Implementação



COISA

ameba

NENHUM TIPO

Ações ▾

- Detalhes
- Segurança**
- Grupos
- Sombra
- Interagir
- Atividades
- Trabalhos

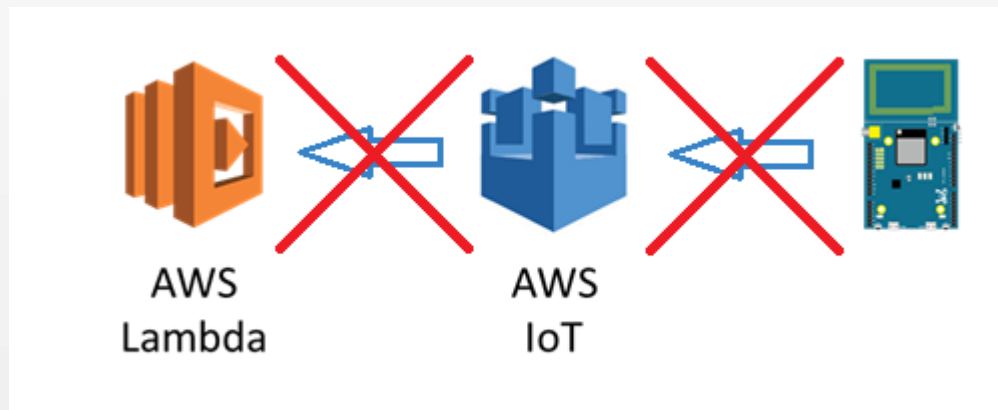
Certificados

[Criar certificado](#) [Visualizar outras opções](#)

6913882ebc553ff53d... ⋮

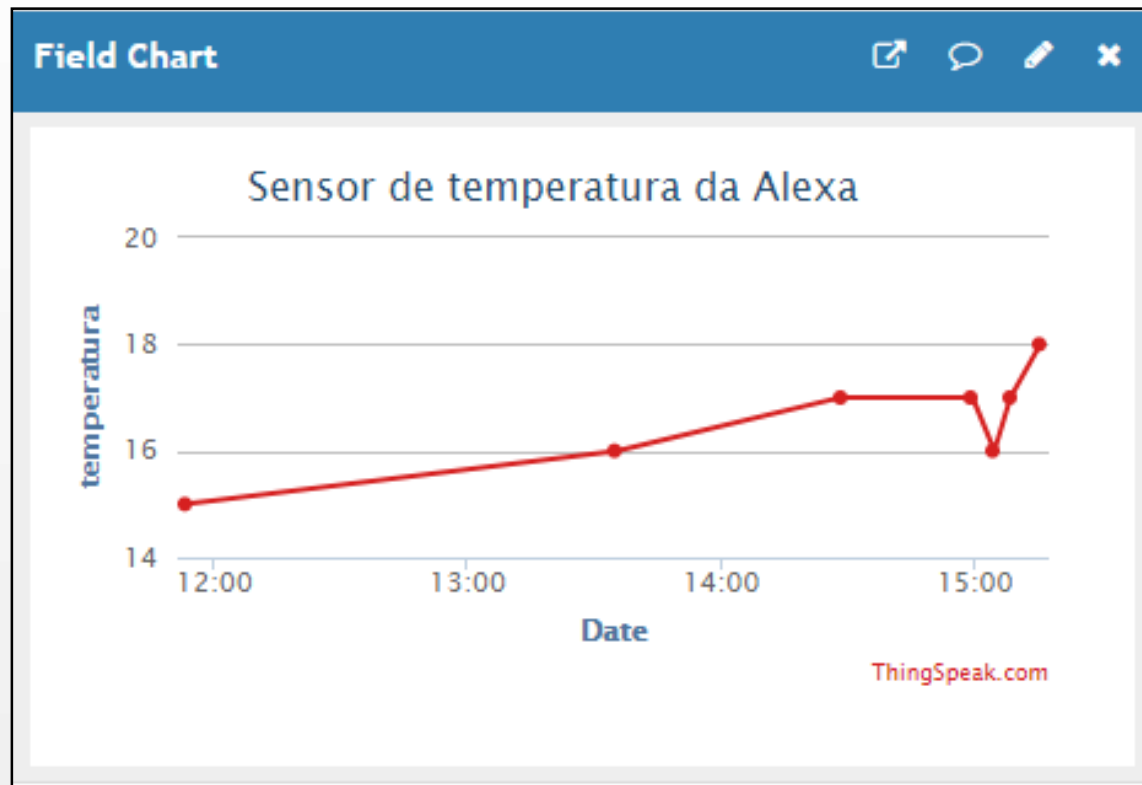
Implementação

- Go MQTT, but don't come back.



Implementação

- HTTP: GET/POST



Implementação

```
309 void sendTemperature(int temperature) {
310     if (thingspeakWifiClient.connect(thingspeakServer,80)) {
311         String postStr = thingspeakApiKey;
312         postStr += "&field1=";
313         postStr += String(temperature);
314
315         thingspeakWifiClient.print("POST /update HTTP/1.1\n");
316         thingspeakWifiClient.print("Host: api.thingspeak.com\n");
317         thingspeakWifiClient.print("Connection: close\n");
318         thingspeakWifiClient.print("X-THINGSPEAKAPIKEY: "+thingspeakApiKey+"\n");
319         thingspeakWifiClient.print("Content-Type: application/x-www-form-urlencoded\n");
320         thingspeakWifiClient.print("Content-Length: ");
321         thingspeakWifiClient.print(postStr.length());
322         thingspeakWifiClient.print("\n\n");
323         thingspeakWifiClient.print(postStr);
324     }
325     thingspeakWifiClient.stop();
326 }
```

Operacionalidade da Implementação

AÇÃO	COMANDO
acender/apagar luz do quarto verde	TELL HOUSE turn the green room light on/off
acender/apagar luz do quarto azul	TELL HOUSE turn the blue room light on/off
acender/apagar luz do banheiro de baixo	TELL HOUSE turn the downstairs bathroom light on/off
acender/apagar luz do banheiro de cima	TELL HOUSE turn the upstairs bathroom light on/off
acender/apagar luz da sala de baixo	TELL HOUSE turn the downstairs livingroom light on/off
acender/apagar luz da sala de cima	TELL HOUSE turn the upstairs livingroom light on/off
acender/apagar luz de fora	TELL HOUSE turn the outside light on/off
acender/apagar luz da cozinha	TELL HOUSE turn the kitchen light on/off
acender/apagar todas as luzes	TELL HOUSE turn all the lights on/off TELL HOUSE turn on/off all the lights TELL HOUSE turn on/off all lights
abrir a porta	TELL HOUSE to open the principal door TELL HOUSE to open the door TELL HOUSE open the door
consultar a temperatura	ASK HOUSE to get the temperature ASK HOUSE to get temperature ASK HOUSE get the temperature ASK HOUSE get temperature

Resultados e Discussões

- Tempo de resposta rápido

FUNCIONALIDADE	TEMPO EM SEGUNDOS
acender ou apagar uma luz	4
responder a temperatura	4
abrir a porta	4

Resultados e Discussões

- Páginas do site da Amazon em constantes atualizações

The screenshot shows the configuration page for an Alexa skill. On the left, there is a sidebar with sections: Skill Information (checked), Interaction Model (checked), Configuration (unchecked), Test (unchecked), Publishing Information (unchecked), and Privacy & Compliance (checked). The main content area has two sections: 'Endpoint' and 'Account Linking'. The 'Endpoint' section has radio buttons for 'HTTPS' (selected) and 'Lambda ARN (Amazon Resource Name)'. Below it is a text input field containing 'arn:aws:lambda:us-east-1:715db2c-ae6f-4f05-b3fc-49b2e8b21a89'. The 'Account Linking' section asks 'Do you allow users to create an account or link to an existing account with you?' with radio buttons for 'Yes' and 'No' (selected). At the bottom, there are three buttons: 'Save', 'Submit for Certification', and 'Next'.

The screenshot shows the configuration page for an Alexa skill, focusing on the 'Endpoint' section. The left sidebar shows the 'Interaction Model' section with 'Intents (5)' expanded to show 'lampada' (with 'Light_state' selected) and 'Built-In Intents (4)' including 'AMAZON_FallbackIntent', 'AMAZON_ConciseIntent', 'AMAZON_HelpIntent', and 'AMAZON_StopIntent'. The 'Slot Types (1)' section shows 'states'. The main content area has a 'Service Endpoint Type' section with radio buttons for 'AWS Lambda ARN' (selected and marked 'Recommended') and 'HTTPS'. Below it, there is a 'Your Skill ID' field with the value 'amzn1.ask.skill.a7f5db2c-ae6f-4f05-b3fc-49b2e8b21a89' and a 'Copy to Clipboard' button. At the bottom, there is a 'Default Region' field with the value 'arn:aws:lambda:us-east-1:609631426702:function:funcap_teste'.

Conclusões e Sugestões

- Dar os comandos de qualquer lugar do mundo a partir de um dispositivo móvel;
- Grande praticidade para dar os comandos;
- Tempo de resposta rápida;
- Em caso de queda de energia o dispositivo se reconecta automaticamente na rede sem fio.

Conclusões e Sugestões

- Abrir a porta usando tag NFC;
- Adicionar novos dispositivos automaticamente;
- Nova forma de representar dispositivos no AWS IoT.