**UNIVERSIDADE REGIONAL DE BLUMENAU**

**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**

**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**JOGO DA MEMÓRIA:   
DESENVOLVIMENTO DE *SKILL* COM CONCEITOS DO JOGO *GENIUS***

**rAFAEL DOS SANTOS RODRIGUES**

**BLUMENAU**

**2021**

**RAFAEL DOS SANTOS RODRIGUES**

**JOGO DA MEMÓRIA:   
DESENVOLVIMENTO DE *SKILL* COM CONCEITOS DO JOGO *GENIUS***

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Dalton Solano dos Reis - Orientador

**BLUMENAU**

**2021**

# Introdução

O ser humano procura a simplificação, delegação ou gerenciamento de suas tarefas pessoais em busca de maior conforto, tempo livre, bem-estar e segurança. De acordo com Costa (2019), vive-se uma verdadeira revolução tecnológica. Constantemente são anunciados novos conceitos, novos desenvolvimentos tecnológicos, novas alusões futuristas, entre outros. A par do momento que se vive, registra-se cada vez mais um aumento na procura, não só num contexto industrial como também num contexto doméstico.

Consequentemente, uma das áreas em destaque da indústria 4.0 seria a *IoT* (*Internet of Things* ou Internet das Coisas). Segundo Nascimento et al (2019), a *IoT* está em rápida evolução como também a rede de periféricos que contém tecnologia de comunicação embarcada. Dessa forma, está cada vez mais comum encontrar dispositivos conectados entre si ou a sistemas externos.

Pode-se citar como uma dessas inovações o surgimento de assistentes pessoais inteligentes, que têm como propósito auxiliar a realizar diversas tarefas do cotidiano, sendo o principal meio de interação os comandos de voz realizados pelo usuário (SILVA, 2021).

A tecnologia de assistentes virtuais não se limita apenas a *smartphones*, já sendo possível ter acesso a outros dispositivos eletrônicos que possuem a tecnologia do comando de voz. Conforme o site Soho (2017) destacou, até alguns anos atrás, os dispositivos domésticos eram limitados ao controle por toque ou controle remoto. Desde o lançamento de dispositivos de automação residencial, como *Amazon Echo* e *Google Home*, a tecnologia de reconhecimento de voz progrediu significativamente e mudou a maneira como interagimos com os aparelhos comuns em casa.

Em suma, Amazon Alexa é um serviço de assistente pessoal inteligente na nuvem que utiliza aprendizagem de máquina e inteligência artificial para realizar diversas ações. Permite solicitar tarefas como pesquisas, criar listas de afazeres, comprar produtos, mandar executar uma lista de músicas ou questionar o horário atual (VIGLIAROLO, 2020). Conforme descrição disponibilizada no site da Amazon Alexa (2021), o serviço permite conectar-se com dispositivos, sejam eles Amazon ou fabricados por terceiros, por meio do Web Service da Amazon (AWS), efetuar comandos de voz, interpretá-los e tomar uma ação correspondente como evocar Application Programming Interfaces (APIs) ou executar uma determinada tarefa.

Um outro exemplo do emprego de tecnologia é o jogo clássico Genius, produzido pela empresa Brinquedos Estrela. Segundo Ferrari (2013) este era um jogo de memória popular na década de 80 que possuía quatro botões coloridos que se iluminavam em sequência e que o objetivo dos jogadores era reproduzir o encadeamento apresentado.

Diante dos fatos apresentados anteriormente, têm-se como propósito estudar a assistente de voz Alexa criando uma *skill* que irá reproduzir algumas das funções do jogo Genius, desenvolvendo uma arquitetura que será composta pela Alexa, controlador inteligente (*ESP32*) e um conjunto de lâmpadas Light Emitting Diodes (LED’s). Isso permitirá um meio de interação alternativa referente ao jogo original aonde a ação do usuário é clicar nas cores corretas com as mãos. Consequentemente será feita uma modernização do jogo clássico que irá abranger toda e qualquer pessoa que tenha interesse em jogar e testar sua capacidade de memorização.

## OBJETIVOS

Este trabalho tem como objetivo geral desenvolver uma *skill* através da plataforma Amazon AWS com arquitetura para integrar lâmpadas LED’s e um controlador inteligente, neste caso o *ESP32,* utilizando o recurso de processamento dos comandos de voz da assistente virtual Alexa. É possível listar como objetivos específicos os itens abaixo:

1. utilizar o recurso da assistente virtual para reconhecer e sintetizar voz, permitindo toda interação do usuário ser exclusivamente por voz, em português brasileiro;
2. ser capaz de tomar ações a partir de um comando do usuário, como por exemplo explicar o objetivo do jogo ou iniciar o jogo;
3. receber uma sequência de palavras por voz e validar se era o valor esperado ou não;
4. integrar a Skill com outros serviços da Amazon;
   1. DynamoDB;
   2. IoT Core;
   3. AWS Lambda;
   4. CloudWatch.
5. Criar um tutorial no idioma português brasileiro de criação e skills;
6. Forma de monetização de uma skill?? – Inserir um tópico sobre isso?
7. controlar o estado (ligado/desligado) da lâmpada através de comandos de voz;
8. Método Relationship of M3C with User Requirements and Usability and Communicability Assessment in groupware (RURUCAg) para analisar e avaliar a usabilidade e a experiência de usuário ao utilizar a *skill*, relacionando os requisitos da aplicação com as heurísticas de Nielsen. à Verificar um método simples

## Estrutura

O trabalho está organizado em cinco capítulos. O primeiro capítulo apresenta a introdução do trabalho desenvolvido, os objetivos a serem alcançados e sua estrutura. No segundo capítulo apresenta os trabalhos correlatos. O terceiro capítulo contém fundamentação teórica para contextualizar este trabalho. Já no quarto capítulo descreve o desenvolvimento do trabalho com os requisitos, as ferramentas utilizadas, a especificação, a implementação, a operacionalidade do protótipo e a análise dos resultados. Por fim, quinto capítulo finaliza com a conclusão do trabalho junto das vantagens e limitações da skill, e as sugestões para trabalhos futuros.

# trabalhos correlatos

A seguir, será feito uma correlação entre três trabalhos com o tema proposto. No item 2.1 deste estudo é detalhado o jogo desenvolvido por Zuffo (2008) que consiste em utilizar como base os conceitos do jogo Genius usufruindo de uma nova interface física. No item 2.2 está descrito o protótipo desenvolvido por Dallarosa Neto (2018), que tem como principal característica o uso da Alexa como intermediária do Arduíno. Por fim, no item 2.3 consta o jogo Pirâmide multiplicativa, um jogo de memória desenvolvido por Rolino, Afini e Vieira (2015).

## Jogo da memória embarcado multinível

Neste trabalho de Zuffo (2008) é realizada uma releitura do jogo Genius da Brinquedos Estrela através do desenvolvimento de um protótipo. O protótipo criado era composto de um microcontrolador Programmable Interrupt Controller (PIC) do microchip responsável pela lógica do jogo, *buzzer* para a emissão de sons, um display para mostrar a sequência, dez botões e quatro cores distintas de LED (Figura 1). O protótipo constava com todas as funcionalidades do jogo Genius original, mas com a diferença no formato passando a ser retangular e o histórico de pontuação com a maior sequência alcançada.

De acordo com Zuffo, a vantagem de usar microcontrolador no lugar de microprocessador é que eles são mais simples e existem vários modelos, facilitando assim para encontrar um microcontrolador com as características ideais para os projetos.

**Figura 1** – Protótipo do jogo

Monitor de computador

Descrição gerada automaticamente com confiança média

Fonte: Zuffo (2008).

O protótipo criado contava com dois modos de jogos, fácil e difícil, tendo como diferença entre eles que o modo fácil contava com um sistema de nova chance, ou seja, mesmo se errar uma sequência era possível continuar de onde errou sem perder a sua pontuação. A cada sequência correta era acrescido um novo elemento a fim de ter um sistema de fases e aumento da dificuldade ao jogador.

O motivo de escolha do tema “jogos” foi pelo fato de que eles possibilitam a criação de ambientes de aprendizagem atraentes e gratificantes, constituindo-se num recurso poderoso de estímulo para o desenvolvimento integral do aluno, permitindo o desenvolvimento de inúmeras habilidades. (ZUFFO)

Para os testes deste equipamento foi utilizado ferramentas específicas para validar todos os periféricos, como o display, led, botões e buzzer. Após a montagem foi disponibilizado para um grupo de 20 pessoas jogarem uma partida em cada nível.

O projeto mostrou-se bastante eficiente com relação ao que foi proposto e o objetivo de desenvolver um jogo de memória eletrônico para estimular a memorização de cores e sons, foi alcançado. (ZUFFO).

Por fim, de acordo com Zuffo, como melhorias futuras, são propostas alterações na alimentação do *buzzer* de forma que permita proporcionar sons mais altos e colocá-lo na parte externa da caixa, reduzir o tamanho da caixa para facilitar o manuseio durante o jogo, definir um tempo para serem pressionados os botões após a exibição da sequência e ajustar a mesma intensidade de luz dos leds para não causar mudanças bruscas que atrapalha na visualização.

## Protótipo de automação residencial utilizando uma assistente de voz

Dallarosa Neto (2018) teve como objetivo criar um protótipo de automação residencial utilizando a assistente virtual Alexa. O protótipo foi criado utilizando para parte de software da assistente virtual a linguagem de programação JavaScript na plataforma Node.js disponibilizado no próprio site da Amazon Web Services. Já na programação do dispositivo embarcado Arduino é utilizada a linguagem de programação C++.

Esse protótipo possibilita alterar e consultar os estados da lâmpada entre ligado/desligado, controlar uma porta eletrônica e consultar a temperatura do ambiente por meio de comandos de voz pelo aplicativo Alexa.

Na Figura 2 é possível observar o processo de como é realizado o comando de voz até a execução da ação no Arduino. Consequentemente, é possível observar que existem quatro camadas: (i) usuário que requisita o comando, (ii) o aplicativo Alexa que interpreta e envia o comando aonde posteriormente retorna uma mensagem de erro ou sucesso ao usuário, (iii) AWS que processa o comando e retorna à Alexa ou envia a ação ao Arduino, e por fim (iv) o Arduino que executa a ação solicitada, assim retornando à AWS resultado de sucesso ou não.

**Figura 2** – Diagrama de Atividades

Diagrama

Descrição gerada automaticamente

Fonte: Dallarosa Neto (2018).

O processo inicia-se pelo comando de voz efetuado pelo usuário. A Alexa sintetiza o comando em formato JSON para o servidor AWS tomar a ação de acordo com o comando. Se a ação solicitada for a temperatura, o AWS consulta a temperatura no servidor ThingSpeak e retorna para a Alexa com o valor da temperatura. Se for um comando para alterar o estado da luz ou abrir a porta, o AWS envia a requisição para o Arduino efetuar a ação correspondente. Além disso, constantemente o Arduino efetua a ação de enviar ao servidor ThingSpeak a temperatura, sem necessidade de uma solicitação do AWS (NETO).

De acordo com Dallarosa Neto, um dos maiores desafios a realizar o desenvolvimento deste protótipo seria a documentação, consequentemente a sua configuração, de como criar o ambiente da Amazon Alexa e do AWS Lambda (2018), das constantes melhorias do ambiente de desenvolvimento do console do desenvolvedor da Alexa e o entendimento do passo-a-passo da Realtek Iot/Arduino Solution que está em idioma estrangeiro, inglês.

Como resultados após testes dessa aplicação a Amazon Alexa foi clara ao falar e assertiva em identificar os comandos do usuário com um baixo tempo de resposta, em torno de quatro segundos, podendo variar dependendo da velocidade de conexão da Internet. Não foram feitos testes com usuários apenas realizado os comandos previstos se executavam o seu propósito.

## AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO

O trabalho de Junior, Javier, Ribeiro e Assis (2020) tem como objetivo desenvolver uma maquete que será embutido um circuito que, conectado à internet por meio da rede Wi-Fi, seja capaz de monitorar a umidade e temperatura do ambiente e controlar, via smartphone, três lâmpadas, uma ventoinha e um servo motor por meio de toques na tela e por comandos de voz.

O modelo escolhido da maquete, segundo Junior, Javier, Ribeiro e Assis (2020), foi inspirado em uma casa moderna tipo sobrado já projetada para projetos dessa natureza, pois ela conta com um compartimento situado no telhado para que seja instalada toda a parte eletrônica incluindo a Protoboard, a placa NodeMCU, o sensor DHT11, o módulo relé e todo o cabeamento que se distribui pelo teto de forma mais organizada.

Abaixo representação da maquete citada anteriormente:

**Figura 3** – Circuito

Desenho de uma caixa

Descrição gerada automaticamente com confiança baixa

Fonte: Junior, Javier, Ribeiro e Assis (2020).

A montagem do circuito consiste nos itens listados a seguir: um NodeMCU, um módulo relé com 4 canais, um servo motor 9g SG90, um sensor de temperatura e umidade DHT11, um protoboard, três interruptores e três lâmpadas. maquete

Abaixo representação do circuito citado anteriormente:

**Figura 4** – Circuito

Diagrama, Esquemático

Descrição gerada automaticamente

Fonte: Junior, Javier, Ribeiro e Assis (2020).

Para a programação do ESP8266 foi utilizado a IDE do Arduino sendo carregada as bibliotecas do “Blynk” para o controle via smartphone, “DHT” para o envio de dados do sensor DHT11 e “Servo” para o controle do servo motor 9g SG90. Para a implementação do controle de voz foi utilizado o serviço “If This Then That” ou “Se Isso Então Aquilo” (Tradução nossa) sendo chamado de IFTTT que segundo Junior, Javier, Ribeiro e Assis (2015) com ele é possível criar instruções condicionais chamadas applets e unir diferentes plicativos da web como o Google e o Facebook em um simples comando sendo utilizado para este projeto uma estrutura condicional para que por meio de um determinado comando de voz feito no Google Assistant, seja acionado um botão no aplicativo Blynk e ative o relé desejado.

O projeto teve funcionamento conforme o esperado, tornando possível realizar as ações citadas anteriormente como conectar o dispositivo via Wi-Fi, monitorar a temperatura e umidade do ambiente, acender e apagar todas as lâmpadas, abrir e fechar o portão da garagem e ligar e desligar a ventilação, tudo isso feito através do smartphone ou por comando de voz feitos ao Google Assistant. Como aprimoramentos futuros, foi citado a adoção de medidas com o foco em segurança, tanto física quanto digital, ou seja, para garantir a segurança física, um sistema de detecção de vazamento de gás por exemplo, e para segurança digital, um sistema de automação imune a hackers pois, como os dados da residência são armazenados na internet, possíveis invasões podem ocorrer, fazendo com que dados importantes sejam roubados. Portanto, para evitar esse tipo de transtorno, é importante que o sistema de automação seja protegido (Junior, Javier, Ribeiro e Assis, 2020).

# REVISÃO BIBLIOGRÁFICA

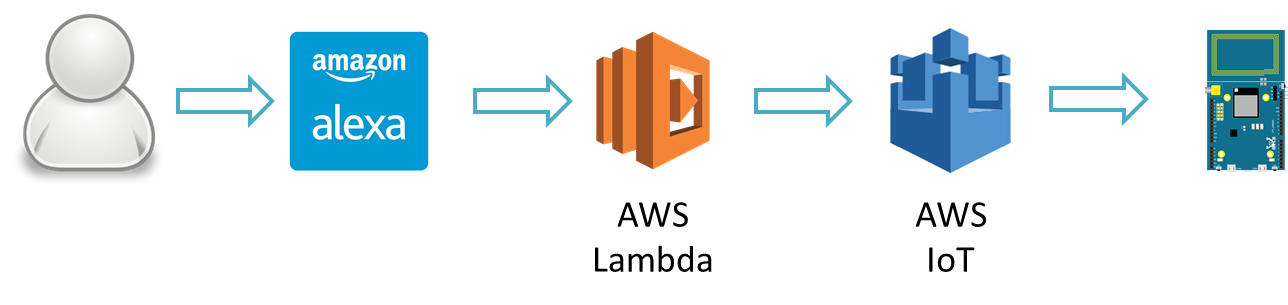
Este capítulo tem por objetivo apresentar os principais assuntos que estão relacionados com o trabalho proposto. A seção 3.1 abordará uma visão geral sobre a assistente virtual Alexa. A seção 3.2 tratará de apresentar o que é a Amazon Web Services e os serviços utilizados, na seção 3.3 explica o jogo Genius e de onde surgiu a ideia, e pôr fim a seção 3.4 apresentará o que é o ESP32 utilizado como um controlador inteligente.

## AMAZON ALEXA

A Alexa é o serviço de voz baseado em nuvem da Amazon disponível em dezenas de milhões de dispositivos da Amazon e de fabricantes de dispositivos de terceiros (AMAZON ALEXA, 2021). Com ela é possível realizar diversas tarefas por comando de voz, por exemplo, receber as principais notícias do dia, qual será a previsão do tempo, lembrete de algum compromisso, comando para desligar/ligar a luz, entre diversos outros, além de conseguir criar a sua própria *skill*.

O nome *skill*, que em tradução literal é habilidade, consiste em um comando de voz a ser interpretado e executado nos servidores da própria Amazon para realizar determinada tarefa. Por esta razão é possível comparar as *skills* da Alexa com os aplicativos utilizados no celular. A Figura 4 apresenta a sequência de passos do funcionamento ao realizar um comando por voz de uma *skill*.

**Figura 4** – Imagem do processamento do comando de voz na Alexa



Fonte: Realtek (2021).

## Amazon Web services

A computação em nuvem consiste em disponibilizar recursos de forma online e sob demanda, pagando apenas aquilo que realmente utilizou não sendo necessário manter de forma física servidores e *datacenters*.

De acordo com a Amazon (AMAZON AWS, 2021), a Amazon Web Services é uma plataforma de serviços de computação em nuvem prestando soluções para empresas, como processamento e armazenamento de dados. O desenvolvedor/cliente que utiliza a tecnologia deve apenas se preocupar em gerenciar o seu negócio ou desenvolver a sua *skill* sem pensar na parte do hardware que está conectado à rede.

O tópico 3.2.1 será apresentado os principais conceitos ao desenvolver uma skill para Alexa, a partir do 3.2.2 ao 3.2.5 será listado os serviços da Amazon utilizados no desenvolvimento da s*kill.*

### Conceitos de desenvolvimento da Skill

Nesta subseção será listado os principais conceitos ao se criar uma skill para a Alexa, ao entender esses conceitos a pessoa já estará apta para criar uma skill única para ela.

COLOCAR IMAGENS DE COMO CRIAR CADA ITEM AQUI?

#### Invocation Name

Invocation name (em tradução livre 'nome de invocação') é a palavra ou frase usada para acionar sua habilidade. De certa forma, é o equivalente da voz a um ícone de aplicativo. Esse nome de invocação geralmente corresponde ao nome da sua habilidade, mas dadas as regras sobre a escolha de um nome de invocação, poderá ser sendo um pouco diferente (ALEXA DEVELOPERS, 2020).

#### Intent

Intent (em tradução livre ‘intenção’) é o que um usuário está tentando realizar. Dentro do código, é o método a ser executado. 'Intenção' não se relaciona com as palavras específicas que um usuário diz, mas com o objetivo de alto nível que ele almeja (ALEXA DEVELOPERS, 2020).

#### Utterance

Utterance (em tradução livre ‘declarações’) são as frases específicas que os usuários usarão ao fazer uma solicitação após abertura da skill e que por consequência executará uma determinada ‘Intenção’. Muitas vezes, há uma grande variedade de enunciados que se encaixam na mesma intenção. E às vezes pode até ser um pouco mais variável (ALEXA DEVELOPERS, 2020).

#### Slots

Um slot é uma variável relacionada a uma intenção que permite ao Alexa entender as informações sobre a solicitação. Por exemplo, em uma skill que sugere um filme aleatório o usuário poderá falar qual gênero do filme ele quer assistir, sendo assim o gênero é a variável que o sistema precisa interpretar de forma não fixa (ALEXA DEVELOPERS, 2020).

##### Tipos de slots

A Amazon fornece vários tipos de slots integrados, como datas, números, durações, hora etc. Mas os desenvolvedores podem criar slots personalizados para variáveis específicas de sua skill (ALEXA DEVELOPERS, 2020).

#### Built-in Intents

Built-in Intents (em tradução livre ‘intenção construída’) são intenções já desenvolvidas e treinadas pelos desenvolvedores da Alexa para serem utilizadas em sua skill, sendo que toda skill é recomendado existir ao menos as intenções: AMAZON.CancelIntent; AMAZON.HelpIntent; AMAZON.StopIntent; AMAZON.RepeatIntent (ALEXA DEVELOPERS, 2020).

#### Alexa Presentation Language

A Alexa tem uma estrutura de design visual chamada Alexa Presentation Language (APL), que permite criar experiências interativas de voz e visual utilizando a tela do dispositivo. APL fornece elementos visuais, incluindo: gráficos, imagens, apresentações de slides e vídeo.

Com ela é possível criar elementos visuais personalizados para dispositivos padrão habilitados para Alexa, como Echo Show, Echo Spot, Fire TV e dispositivos Fire Tablet selecionados. Dispositivos de terceiros criados usando o Alexa Smart Screen e o SDK do dispositivo de TV também suportam a estrutura de design APL (ALEXA DEVELOPERS, 2020a).

### AWS Lambda

Um dos serviços disponibilizados é o AWS Lambda, que segundo a Amazon (AMAZON LAMBDA, 2021) o serviço permite executar o código sem provisionar ou gerenciar servidores, onde você paga apenas pelo tempo efetivo de computação que utilizar, ou seja, apenas quando existir o processamento ativo nos servidores e quando o código não estiver em execução não será cobrado nenhum valor. O serviço permite executar código ou serviço de *backend*, para isso se deve realizar o *upload* do código e o Lambda irá se encarregar dos itens necessários para execução. É possível fazer com que ele seja acionado automaticamente por meio de outros produtos da AWS ou chamá-lo diretamente usando qualquer aplicação móvel ou da Web.

-- Devo colocar a logo do Lambda?

### DynamoDB

Para a persistência de dados após o uso da skill é utilizado o serviço DynamoDB, segundo ZZZZ DynamoDB é um banco de dados de chave-valor NoSQL, sem servidor e totalmente gerenciado, projetado para executar aplicações de alta performance em qualquer escala. O DynamoDB como característica têm segurança integrada, backups contínuos, replicação multirregional automatizada, armazenamento em cache na memória e ferramentas de importação e exportação de dados.

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html> -- Não sei como colocar o autor pois não posso usar Amazon novamente. Devo colocar a logo do DynamoDB?

### CloudWatch

Para auxílio do desenvolvimento da *skill* é vinculado o serviço *CloudWatch,* comeste serviço é possível gerar logs e acompanhar em tempo real a execução da sua *skill.* Outras características é possível citar a criação de alarmes, coletar e monitorar métricas de suas aplicações e criação de painéis com as métricas escolhidas. (ZZZZ,2022).

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html> -- Não sei como colocar o autor pois não posso usar Amazon novamente. Devo colocar a logo do CloudWatch?

### IoT Core

AWS IoT Core é também um gerenciamento na nuvem que permite a conexão com outros aparelhos e serviços. Segundo ZZZZ, o AWS IoT Core pode oferecer suporte a bilhões de dispositivos e trilhões de mensagens e pode processar e rotear essas mensagens para endpoints da AWS e outros dispositivos de maneira confiável e segura utilizando os protocolos MQTT ou HTTPS.

<https://docs.aws.amazon.com/whitepapers/latest/aws-overview/internet-of-things-services.html#aws-iot-core/> -- Não sei como colocar o autor pois não posso usar Amazon novamente. Devo colocar a logo do Iot Core?

## GENIUS

O jogo Genius era um brinquedo muito popular na década de 80 no Brasil, e que é comercializado até hoje. Lançado pela Brinquedos Estrela SA o brinquedo buscava estimular a memorização de cores e sons. Utilizava um formato simples e oval e possuía botões coloridos (Figura 5) que emitiam sons harmônicos e se iluminavam em sequência.

O objetivo deste jogo é reproduzir a sequência de cores apresentada previamente sem errar, contando com três fases distintas, cuja diferença é marcada pela velocidade que as cores são apresentadas, com o tempo médio de jogatina por 15 minutos por jogo.

O motivo do jogo Genius ter sido escolhido se deve ao fato de ele se tratar de um jogo que estimula a memória, e esta tem um papel importante no dia a dia do homem e com o avanço da tecnologia, como aponta Almeida (2021), existem cada vez mais comodidades fazendo com que não seja desenvolvida a capacidade cognitiva. Segundo Gunter (2020), com o jogo da memória são obtidos benefícios como a de desenvolver habilidades de concentração, autonomia e confiança. –Retirar?

**Figura 5** – Jogo Genius



Fonte: VejaSP (2017).

## ESP32

Desenvolvido pela empresa Espressif, o ESP32 é um dispositivo IoT (Internet das Coisas) que consiste em um microprocessador de baixa potência dual core com suporte embutido à rede WiFi, Bluetooth e memória flash integrada. Essa arquitetura permite que ele possa ser programado de forma independente, sem a necessidade de outras placas microcontroladoras como o Arduino, por exemplo. Dentre as principais características deste dispositivo, pode-se citar: baixo consumo de energia, alto desempenho de potência, amplificador de baixo ruído, robustez, versatilidade e confiabilidade. (OLIVEIRA, 2017).

Segundo Locatelli (2018), umas das alternativas de software mais utilizadas no desenvolvimento inicial de programações em placas ESP tem sido a plataforma de desenvolvimento Arduino, pois, este possui uma grande comunidade de desenvolvedores, que costumam disponibilizar diversos exemplos e projetos de forma *Open Source* (gratuita).

O intuito é transformar o ESP32 em um controlador inteligente.

Interface gráfica do usuário

Descrição gerada automaticamente

### Controlador inteligente

Um controlador inteligente é um interruptor tradicional que possui boa capacidade de processamento e fácil integração com demais computadores. O intuito dele é gerenciar os aparelhos a ele conectado, sendo usualmente lâmpadas, com isso ele pode além de gerenciar o acionamento das luzes, também permite controlar a intensidade. Permite também criar uma política de horários para acender ou apagar em momentos pré-determinados e controlar a sua situação também por meio de aplicativos. (SAGE, 2021).

O controlador inteligente pode ser manipulado via aplicativo específico ou por intermédio de uma assistente virtual, que neste projeto será utilizada a assistente virtual Alexa. Desta forma, ao vincularmos a Alexa neste componente é possível realizar os comandos por voz pretendendo tornar a experiência mais dinâmica, simples e conveniente. Usualmente, é feito o comando direto por voz para ligar e desligar a lâmpada, mas neste caso será executado o jogo e ele irá definir qual será a ação da controladora.

# Desenvolvimento da Skill

No presente capítulo são descritas as etapas para o desenvolvimento deste projeto e é abordado resumidamente as ferramentas utilizadas. Na o segmento 4.1 contém os requisitos funcionais e não funcionais, na seção 4.2 contém a especificação detalhada do funcionamento da skill, já na seção 4.3 tem detalhes da implementação, os principais conceitos ao desenvolver uma skill e a montagem da maquete de demonstração e na seção 4.4 é feita uma análise dos resultados obtidos da skill.



## Levantamento de requisitos

Este trabalho apresenta o desenvolvimento de uma skill utilizada na assistente virtual Alexa que conecta via protocolo MQTT com o componente ESP32 para ligar e desligar lâmpadas com o objetivo de se criar um jogo. A skill é composta por 6 principais partes: processamento do comando de voz do usuário, envio do JSON criado para o servidor Lambda, execução do código fonte baseado nas informações do JSON, utilização do protocolo MQTT para comunicação com o ESP32, salvar os dados na sessão, salvar os dados no banco de dados.

## Especificação

Nesta seção são apresentadas as especificações técnicas das aplicações e os diagramas deste trabalho. Para tal, os requisitos são apresentados na subseção 4.2.1, contendo as Regras de Negócio (RN), os Requisitos Funcionais (RF) e os Requisitos Não Funcionais (RNF). Na subseção 4.2.2 é apresentado o Diagrama de Caso de Uso (DCU) criado na ferramenta draw.io, subseção 4.2.3 é exposta a matriz de rastreabilidade entre os Casos de Uso (CU) e sua relação com os RF, na subseção 4.2.4 é exibida o diagrama de atividades também criado na ferramenta draw.io. Por fim, é demonstrado o diagrama esquemático na subseção 4.2.5 feito na ferramenta circuito.io.

### Requisitos

Esta subseção apresenta as Regras de Negócio (RN) no Quadro 1, os Requisitos Funcionais (RF) no Quadro 2 e os Requisitos Não Funcionais (RNF) no Quadro 3, da skill desenvolvida.

Quadro 1 - Regras de negócio da skill

|  |  |
| --- | --- |
| RN | DESCRIÇÃO |
| RN01 | O usuário só poderá salvar os seus dados caso a sua pontuação seja maior do que a anterior. |
| RN02 | O usuário só poderá responder à questão após o comando. |
| RN03 | O usuário só poderá responder com cores (pré-definidas) ou números dependendo da pergunta. |

Quadro 2 - Requisitos Funcionais da skill

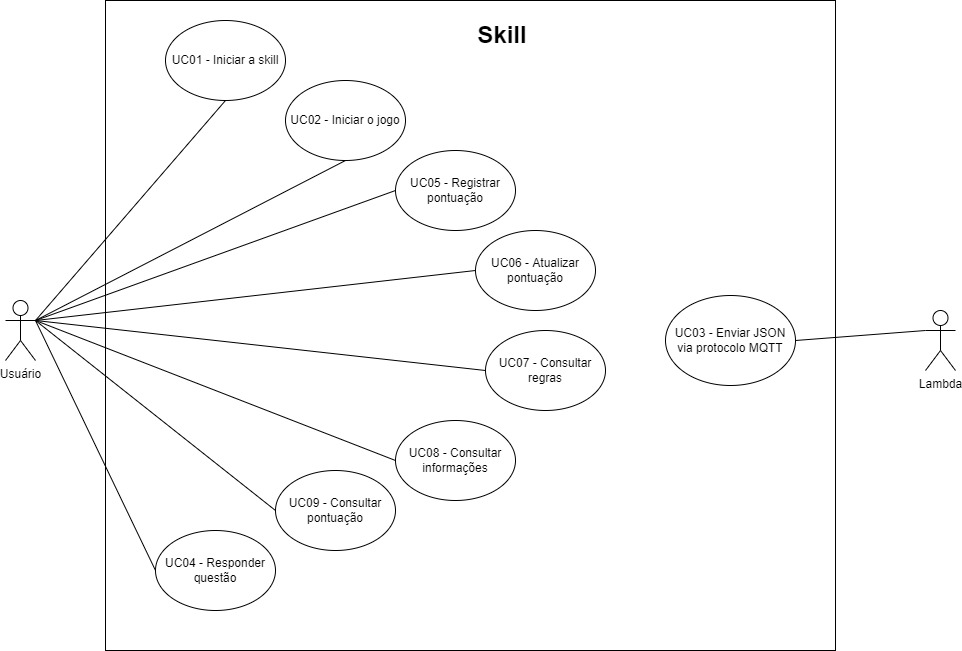
|  |  |
| --- | --- |
| RF | O sistema deverá: |
| RF01 | reconhecer comandos de voz pré-definidos. |
| RF02 | usar comando de voz para invocar a skill. |
| RF03 | permitir escolher a dificuldade. |
| RF04 | gravar na sessão a soma da pontuação. |
| RF05 | gravar no banco de dados a maior pontuação e o nome do usuário. |
| RF06 | conferir a resposta por voz do usuário. |
| RF07 | acionar (ligar/desligar) a lâmpada. |

Quadro 3- Requisitos Não funcionais da skill

|  |  |
| --- | --- |
| RNF | O sistema deverá: |
| RNF01 | atualizar a maior pontuação do usuário no banco. |
| RNF02 | ser desenvolvido em Python e na própria plataforma disponibilizada pela Amazon. |
| RNF03 | utilizar a plataforma da Amazon Web Services (AWS). |
| RNF04 | utilizar o serviço CloudWatch da AWS. |
| RNF05 | utilizar o serviço DynamoDB da AWS. |
| RNF06 | utilizar o serviço IoT Core da AWS. |
| RNF07 | utilizar o serviço Lambda da AWS. |
| RNF08 | utilizar o protocolo MQTT. |
| RNF09 | ser desenvolvido na plataforma Arduino Ameba. |
| RNF10 | utilizar lâmpadas e uma controladora na sua arquitetura. |

### Diagrama de Casos de Uso

Nesta subseção é apresentado o diagrama de casos de uso que mostras as funcionalidades da aplicação feita pelo usuário e servidor. A explicação do diagrama de casos de uso está no Apêndice A. A Figura X apresenta o diagrama de casos de uso das funcionalidades do protótipo.



No caso de uso UC01 o ator efetua um comando de voz para iniciar a skill. O UC02 representa o comando para iniciar o jogo. O UC03 sendo o servidor que após conexão pública o JSON para o tópico do ESP32. O UC04 para responder à questão proposta. O UC05 para registrar a pontuação após terminar uma partida. O UC06 para atualizar a pontuação que já existe deste jogador. O UC07 para consultar as regras. O UC08 para consultar as informações. E O UC09 para consultar a lista de pontuação.

### Matriz de rastreabilidade dos RF e sua relação com os Casos de Uso

Esta subseção apresenta o Quadro X, que exibe a matriz de rastreabilidade dos RF com os UCs referente a skill desenvolvida.

|  |  |
| --- | --- |
| **RF** |  |
| RF01 | UC01, UC02, UC04, UC05, UC06, UC07, UC08, UC09. |
| RF02 | UC01. |
| RF03 | UC02. |
| RF04 | UC05. |
| RF05 | UC06. |
| RF06 | UC04. |
| RF07 | UC03. |

### Diagrama de atividades

O diagrama de atividades da Figura X apresenta os fluxos de execução da skill de acordo com a ação de iniciar o jogo escolhendo o modo e dificuldade.

O processo inicia-se pelo comando de voz efetuado pelo usuário. A Amazon Alexa sintetiza o comando em formato JSON para o servidor AWS tomar a ação de acordo com o comando. Se for um comando para iniciar o jogo, o AWS envia a requisição para o Arduino efetuar a ação correspondente. Enfim, o Arduino após a execução do que foi requisitado retorna para um tópico na Amazon novamente um JSON apenas para ser gravado um log.

Diagrama

Descrição gerada automaticamente

### Diagrama esquemático

Foi desenhado todo o circuito para o protótipo na Figura X utilizando a ferramenta circuito.io. O diagrama mostra o Arduino conectado em uma protoboard, que realiza contém uma conexão com uma resistência e posteriormente com o LED, utilizando o cabo micro USB para conectar em uma fonte de alimentação de 5V ou em uma Powerbank. Também é feita a conexão “terra” entre a protoboard e o Arduino.

Tela de um aparelho eletrônico

Descrição gerada automaticamente com confiança baixa

## Implementação

Nesta seção são descritas as formas de implementação da skill, que segue a seguinte estrutura: a subseção 3.3.1 descreve as técnicas e ferramentas utilizadas, bem como o esquema de tecnologias; a subseção 3.3.2 descreve a codificação da aplicação; e, por fim, a subseção 3.3.3 apresenta a operacionalidade da implementação.

### Técnicas e ferramentas utilizadas

Esta subseção descreve as técnicas e as ferramentas utilizadas na implementação da skill desenvolvida que foram utilizadas em diferentes fases durante a construção deste trabalho. Na parte inicial foi realizado um estudo sobre os temas do trabalho, analisando e aprofundando os assuntos citados na seção de fundamentação teórica (seção X).

Após esta primeira etapa, foi realizada a análise dos requisitos para a construção a skill, estando presentes na seção X. Como resultado, foram desenvolvidos os diagramas de casos de uso, de atividades e esquemático. Os diagramas de uso e esquemático foram elaborados utilizando a linguagem Unified Modeling Language (UML) e fazendo uso da ferramenta Draw.io já o esquemático utilizando a ferramenta circuito.io.

O desenvolvimento da skill foi utilizada a própria plataforma de desenvolvimento da Alexa, Alexa Console Developer, utilizando a linguagem de programação Python e posteriormente exportado esse código para o AWS Lambda. A aplicação embarcada foi desenvolvida utilizando a linguagem de programação C++ utilizando o ambiente de desenvolvimento Arduino IDE.

Como material complementar contendo tutorial, códigos fontes, JSON com as frases da skill e exemplos de testes utilizando os recursos da Amazon foi criado um repositório que pode ser encontrado em Rodrigues (2022).

#### Preparação do circuito eletrônico para a maquete

Para a criação da carcaça do trabalho foi utilizado a carcaça do jogo da memória da empresa Unik Toys aproveitando também a placa de circuitos e retirando outros componentes desnecessários. O projeto físico conta com 4 LEDs para serem utilizados.

Na Figura X pode-se visualizar o circuito antes da montagem com a carcaça, na figura X o circuito dentro da carcaça e por fim na figura X o projeto físico montado.

Em suma, foi utilizado na montagem do projeto os seguintes componentes:

1. 4 resistências de 220ohms custando cada R$0,10,
2. 1 ESP-WROOM-32 DEVKIT V1l WIFI e Bluetooth – NODEMCU que custou R$84,43,
3. 1 Cabo Micro USB que custou R$19,59,
4. Jumper Femea x Femea, kit com 10 unidades, 20cm que custou R$3,90,
5. Fonte de alimentação 5V,
6. 5 Jumper Macho x Macho,
7. Jogo da memória Unik Toys que custou R$ 53,24

O valor total para a realização do projeto foi de R$X.

<https://proesi.com.br/jumper-femea-x-femea-kit-com-10-unidades-12cm.html>  
https://proesi.com.br/esp-wroom-32-devkit-v1l-wifi-e-bluetooth-nodemcu.html  
https://proesi.com.br/cabo-micro-usb-para-usb-a-2-0-1m-preto-pmuap-1.html

https://proesi.com.br/led-difuso-5mm-azul.html

https://proesi.com.br/led-difuso-5mm-vermelho.html

https://proesi.com.br/led-difuso-5mm-verde.html

https://proesi.com.br/led-difuso-5mm-amarelo.html

<https://proesi.com.br/resistor-precisao-1-1-4w-220r.html>

https://www.amazon.com.br/gp/product/B06X96ZSM2/ref=ppx\_yo\_dt\_b\_asin\_title\_o07\_s00?ie=UTF8&psc=1

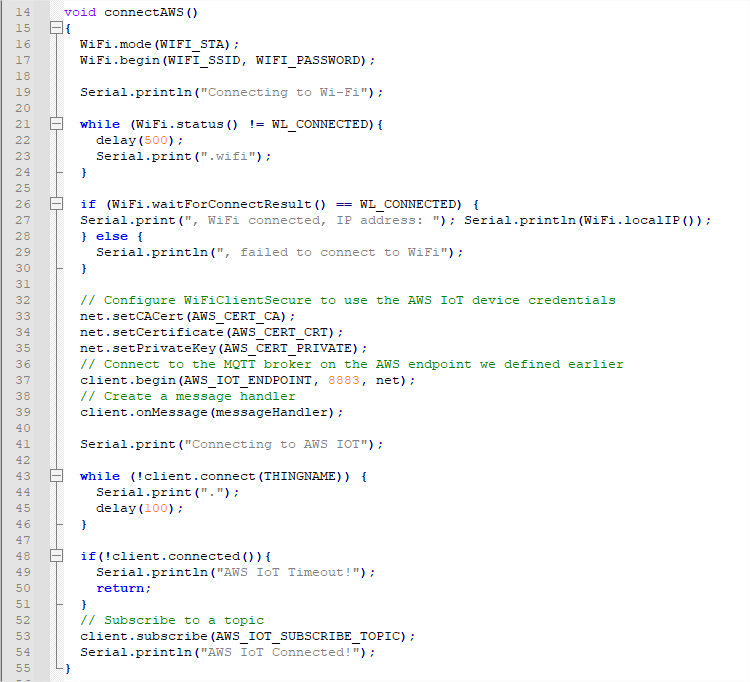
#### Programando o Arduino Ameba

Conforme Rodrigues (2022) os métodos essenciais que deve existir no código do Arduino e consequentemente os primeiros a serem executados seria setup e loop. O setup é executado sempre que o Arduino é iniciado. O loop é um laço que é constantemente executado após ele mesmo terminar a execução do setup. Na Quadro X mostra parte do código do método setup adaptado de um exemplo gerado dentro da Arduino IDE, neste caso é executado dois métodos para tentar se conectar ao Wi-Fi e posteriormente a Amazon e escolher os conectores do ESP32 que serão utilizados.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

No quadro 2 consta o método connectAWS utiliza as bibliotecas WiFi para primeiramente conectar-se à rede e a WiFiClientSecure para utilizar os certificados (33, 34 e 35) gerados na Amazon e conseguir se conectar na “Coisa” e posteriormente se inscrever no tópico.



No quadro X apresenta a rotina messageHandler que irá receber o tópico e o arquivo JSON que foi enviado do servidor da AWS para o Arduino, ele irá basicamente receber o JSON, deserializar o conteúdo para criar uma lista com a sequência das lâmpadas que devem ser ligadas e desligadas. Por exemplo, para acender uma lâmpada é utilizado o método digitalWrite utilizando como segundo parâmetro o valor “HIGH” e para apagar o valor “LOW”, esses valores depende do circuito eletrônico.

Por fim é executado o método publishReturnMessage que irá enviar um novo JSON ao servidor da Amazon para geração de logs.

Obs.: O método publishReturnMessage tem o intuito de apresentar como é feito um envio do Arduino para a AWS assim mostrando ambos os casos de receber e enviar informações entre a AWS e Arduino.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Em Rodrigues (2022) contém maiores informações sobre os métodos, configuração e funcionamento do código fonte utilizado.

#### Estrutura da Amazon Web Services

Para o desenvolvimento deste projeto foi necessário entender e utilizar os conceitos expostos na seção X, sendo que como base foi realizado o curso “Desenvolvimento Alexa - 15 Skills Programadas em Python” disponibilizado na plataforma Udemy e assistido a lista de vídeos “Zero to Hero (em tradução livre ‘Do Zero ao Herói’)” do canal oficial dos desenvolvedores da Alexa que pode ser encontrado em ALEXA DEVELOPERS (2020).

No entanto, por essas ferramentas (Console do desenvolvedor Alexa, AWS Lambda, AWS IoT) estarem em constate desenvolvimentos o tutorial/curso podem estar defasados ou incompletos além de grande parte do conteúdo encontrado na internet ou documentação oficial serem em inglês aumentando assim a dificuldade.

Levando em consideração essas dificuldades em Rodrigues (2022) também conta em como criar a estrutura de uma skill em português brasileiro além de como utilizar os outros serviços de logs, servidor Lambda, IoT e DynamoDB.

Como resumo do que é necessário, primeiramente cria-se uma skill na Amazon Alexa escolhendo o tipo e a linguagem de programação a ser utilizada. Após isso, cadastra-se o nome de invocação, este nome seria o que a Alexa irá reconhecer para abrir a skill criada. Em seguida, são criadas as intenções, e suas frases de ativação, ou seja, as suas declarações que fará a Alexa solicitar ao AWS Lambda que seja executado determinado comando.

As declarações são criadas na tela de intenções dentro do item correspondente, conforme figura X. É possível cadastrar mais de uma frase para ativar uma intenção sendo que o recomendado é ao menos ter 7 formas distintas (ALEXA DEVELOPERS, 2020).

Nas declarações é possível ter o equivalente a variável sendo que o valor será o que o usuário falar e que é aceito apenas dentro de uma lista pré-definida de valores que podemos cadastrar ou utilizar o padrão disponibilizado pela Amazon, esse conceito é o que chamamos de slots. Na figura X consta o slot ‘num’ que aceita apenas números.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Por fim, quando o usuário utilizar uma declaração será feita uma chamada ao AWS Lambda que realizará a execução do método correspondente aquela intenção, por exemplo podendo enviar um JSON ao Arduino por protocolo MQTT, criar um registro no banco de dados ou apenas retornar um texto a ser falado para o usuário.

#### Operacionalidade da implementação

Para a realização dos testes primeiros é necessário configurar a Echo Show para ter acesso a internet, ligar o jogo em uma fonte de 5V ou a uma PowerBank e confirmar que ele tem acesso à internet para estabelecer a conexão com o servidor da AWS Lambda.

O quadro X consta com as principais frases em português brasileito a serem ditas para realizar determinada ação, sendo a palavra envolvida por chave o slot.

|  |  |
| --- | --- |
| AÇÃO | COMANDO |
| Iniciar a skill | Abrir jogo da memória |
| Iniciar o jogo | Dificuldade {difficulty} modo {gametype} |
| Responder à questão de sequência de cores | {CorUm}{CorDois}{CorTres} {CorQuatro} |
| Informações ou regras do jogo | Eu quero mais informações do jogo |
| Registrar ou exclui pontuação | {ação} minha pontuação |
| Lista das pontuações registradas | Mostre a pontuação |
| Receber a questão a ser respondida | Pronto |
| Responder à questão da quantidade de cores | Eu acho que a resposta é {número} |

Já o quadro X consta com as frases em inglês estadunidense.

|  |  |
| --- | --- |
| AÇÃO | COMANDO |
| Iniciar a skill | Open memory genius |
| Iniciar o jogo | I want to play in {difficulty} difficulty and {gametype} mode |
| Responder à questão de sequência de cores | {color sequence} |
| Informações ou regras do jogo | Information about the game |
| Registrar ou exclui pontuação | {action} my points |
| Lista das pontuações registradas | Tell the ranking list |
| Receber a questão a ser respondida | Ready |
| Responder a questão da quantidade de cores | I think the answer is {num} |

Conforme quadros X e Y, é possível verificar que existe uma diferença entre o item “Responder à questão de sequência de cores” pois ao realizar a criação da skill em português brasileiro o idioma não consta com uma configuração de slot “multi-value” que recebe um texto com tudo que o usuário fala.

No apêndice X consta a execução dos comandos no console do desenvolvedor da Alexa e o retorno esperado de cada comando.

## Análise de resultados

# Conclusões

## Conclusões

Referências

ALMEIDA, JÉSSICA. **O uso de tecnologia afeta a nossa memória?** Rio de Janeiro, 2021. Disponível em: https://www.appai.org.br/appai-educacao-revista-appai-educar-edicao-125-o-uso-de-tecnologia-afeta-a-nossa-memoria/. Acesso em 06 de set. 2021.

AMAZON ALEXA. **Amazon Alexa**, 2021. Disponível em: https://developer.amazon.com/en-US/alexa. Acesso em: 28 ago. 2021.

AMAZON AWS. **Computação em nuvem com a AWS**,2021. Disponível em: https://aws.amazon.com/pt/what-is-aws/. Acesso em: 06 de set. 2021.

AMAZON LAMBDA. **AWS Lambda**, 2021. Disponível em: https://aws.amazon.com/pt/lambda/. Acesso em: 29 ago. 2021.

BRENELLI, R. P. **O Jogo como Espaço para Pensar:** A Construção de Noções Lógicas e Aritméticas. Editora Papirus. 2000.

DALAROSSA NETO, Leandro. **Protótipo de automação residencial utilizando uma assistente de voz.** 2018. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FERRARI, Rafael. **GENIUS -** O clássico, 2013. Disponível em:http://rafa.eng.br/genius.htm. Acesso em: 25 ago. 2021.

GUNTER, Ricardo. **Um jogo para todas as disciplinas**, 2020. Disponível em:https://www.appai.org.br/appai-educacao-revista-appai-educar-edicao-122-um-jogo-para-todas-as-disciplinas/. Acesso em: 03 out. 2021.

REALTEK. **Ameba Arduino:** [RTL8195AM] Amazon Alexa, 2012. Disponível em: https://www.amebaiot.com/en/ameba-arduino-amazon-alexa/. Acesso em: 28 ago. 2021.

VejaSP. **Dez brinquedos inesquecíveis dos anos 80**. São Paulo, 27 de fev. de 2017. Disponível em: https://vejasp.abril.com.br/blog/vejinha/dez-brinquedos-inesqueciveis-dos-anos-80/. Acesso em: 01 de set. de 2021.

VICENTE, MARCOS. **GENIUS.** São Paulo, [2021?]. Disponível em: http://www.autobahn.com.br/brinquedos/genius.html. Acesso em: 06 de set. 2021.

VIGLIAROLO, Brandon. **Amazon Alexa:** Cheat Sheet. [S.I.], 2020. Disponível em: https://www.techrepublic.com/article/amazon-alexa-the-smart-persons-guide. Acesso em: 28 ago. 2021.

ZUFFO, Eduardo H. **Jogo da memória embarcado multinível.** 2008. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) - Núcleo de Ciências Exatas e Tecnológicas, Universidade Positivo, Curitiba.

# Apêndices

Este apêndice descreve os casos de uso abordados nos diagramas expostos na seção 3.2.1 deste trabalho. No Quadro X apresenta-se a descrição do caso de uso Iniciar a skill.

Quadro X4 - Descrição do UC01 Iniciar a skill

|  |
| --- |
| UC–01 – Iniciar a skill  Iniciar a skill após o usuário utilizar o nome de invocação.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pós-condição: Alexa inicia a skill solicitada.  Cenário Principal:  1. Usuário utiliza nome de invocação;  2. A Alexa reconhece o nome de invocação;  3. Alexa inicia uma nova sessão para o usuário. |

Quadro X - Descrição do UC02 Iniciar o jogo

|  |
| --- |
| UC–02 – Iniciar a skill  Iniciar o jogo após o usuário utilizar o comando contendo os parâmetros do jogo.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pré-condição: Alexa já está com uma sessão em aberta.  Pós-condição: Alexa inicia o jogo.  Cenário Principal:  1. Usuário utiliza o comando para iniciar o jogo passando os parâmetros de dificuldade e modo de jogo;  2. A Alexa reconhece o comando;  3. Alexa cria uma sequência de cores e faz uma requisição MQTT. |

Quadro X - Descrição do UC03 Enviar JSON via protocolo MQTT

|  |
| --- |
| UC03 – Enviar JSON via protocolo MQTT  Enviar JSON via protocolo MQTT ao Arduino.  Ator: Lambda  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pré-condição: UC01 e UC02.  Pós-condição: Arduino liga/desliga as lâmpadas baseado no comando recebido.  Cenário Principal:  1. Lambda conecta no tópico MQTT;  2. Lambda publica um JSON no tópico inscrito;  3. Arduino realiza o comando recebido. |

Quadro X - Descrição do UC04 Registrar pontuação

|  |
| --- |
| UC04 – Responde questão  Registrar a pontuação da sessão daquele usuário.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pré-condição: UC01, UC02 e UC03.  Pós-condição: Alexa retorna sucesso ou erro ao usuário.  Cenário Principal:  1. Usuário utiliza o comando para receber a questão;  2. A Alexa reconhece o comando e retorna uma nova questão;  3. Usuário responde à questão;  4. Alexa retorna uma mensagem de sucesso ou erro. |

Quadro X - Descrição do UC05 Registrar pontuação

|  |
| --- |
| UC05 – Registrar pontuação  Registrar a pontuação da sessão daquele usuário.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pré-condição: UC01, UC02 e UC03.  Pós-condição: Alexa retorna sucesso ou erro ao usuário.  Cenário Principal:  1. Usuário utiliza o comando para registrar a sua pontuação passando, nome e sobrenome como parâmetros;  2. A Alexa reconhece o comando e valida se já não existe o registro do usuário;  3. Alexa cria um registro para o usuário com a pontuação da sessão. |

Quadro X - Descrição do UC06 Atualizar pontuação

|  |
| --- |
| UC06 – Atualizar pontuação  Atualizar a pontuação do registro do jogador existente na base de dados.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pré-condição: UC01, UC02, UC04 e UC05.  Pós-condição: Alexa atualiza o registro do usuário.  Cenário Principal:  1. Usuário utiliza o comando para registrar a sua pontuação passando, nome e sobrenome como parâmetros;  2. A Alexa reconhece o comando e valida se já não existe o registro do usuário;  3. Alexa atualiza o registro daquele usuário com a pontuação da sessão. |

Quadro X - Descrição do UC07 Consultar regras

|  |
| --- |
| UC07 – Consultar regras  Alexa retorna uma frase com as regras do jogo.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pós-condição: Alexa retorna uma frase com as regras do jogo.  Cenário Principal:  1. Usuário utiliza o comando para requisitar as regras do jogo;  2. A Alexa reconhece o comando;  3. Alexa fala as regras do jogo. |

Quadro X - Descrição do UC08 Consultar Informações

|  |
| --- |
| UC08 – Consultar regras  Alexa retorna uma frase com a informação do jogo.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pós-condição: Alexa retorna uma frase com a informação do jogo.  Cenário Principal:  1. Usuário utiliza o comando para requisitar informações do jogo;  2. A Alexa reconhece o comando;  3. Alexa fala as informações do jogo. |

Quadro X - Descrição do UC09 Consultar Informações

|  |
| --- |
| UC09– Consultar regras  Consultar a lista de pontuação.  Ator: Usuário  Pré-condição: A Alexa deve estar ligada.  Pré-condição: A Alexa deve estar conectada na internet.  Pós-condição: Alexa retorna uma frase com a lista de pontuação do jogo.  Cenário Principal:  1. Usuário utiliza o comando para requisitar a lista de pontuação;  2. A Alexa reconhece o comando;  3. Alexa fala a lista de pontuação dos três primeiros colocados. |

Como argumento técnico destaca-se o estudo sobre o uso dos recursos das assistentes virtuais, bem como das plataformas online de desenvolvimento onde será necessário utilizar a linguagem Python juntamente com a plataforma Amazon Web Services. Como contribuição prática ou social ressalta-se a sua aplicação para todos, tornando jogos simples e até mesmo educacionais mais dinâmicos e interativos, pois permitirá a utilização da fala para controlar e jogar. Observa-se que foram encontrados aplicativos similares na loja de skills da Alexa mas nenhum deles foi possível testar pois são bloqueados geograficamente mas pela descrições dos jogos nenhum utiliza um dispositivo externo a Alexa para auxiliar.