

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
2022

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
2022**

Esta página deverá ser substituída pela folha
de assinaturas entregue na Banca.

Digitalize a folha e cole aqui para a entrega da
versão final do TCC.

Atenção: não ultrapasse as margens!

Dedico este trabalho aos meus familiares e a todas as pessoas que me ajudaram nesta trajetória.

AGRADECIMENTOS


Primeiramente gostaria de agradecer ao meus pais que sempre me apoiaram e incentivaram a estudar, sendo a minha base para tudo mesmo em seus momentos frágeis em que pensei em desistir nessa etapa final. A minha irmã por sempre se preocupar com meu bem-estar e aos amigos verdadeiros que consegui nesse percurso.

A todas as pessoas que contribuíram de alguma forma para a realização deste trabalho, fornecendo ideias, correção do texto e pontos de melhorias.

Por fim, gostaria de agradecer de forma especial ao professor Dalton Solano Reis pela confiança, apoio e orientação para poder concluir este trabalho.

Não importa o quão devagar você vá, desde
que você não pare.
Confúcio

RESUMO



Este trabalho apresenta o levantamento de requisitos, a especificação, o desenvolvimento e a operacionalidade de uma *skill* para a assistente virtual Alexa. O objetivo principal deste trabalho é desenvolver uma *skill* utilizando a plataforma da Amazon Web Services, no qual será uma releitura do jogo clássico Genius. Um dos objetivos específicos é utilizar o recurso da assistente virtual para reconhecer e sintetizar voz para realizar toda a interação entre o usuário e o jogo. A parte de interação da *skill* foi desenvolvida utilizando o console do desenvolvedor da Alexa, a lógica da *skill* foi desenvolvida na plataforma Amazon Web Service Lambda utilizando a linguagem de programação Python. A *skill* se comunica com o microcontrolador Arduino ESP32 utilizando o protocolo de comunicação Message Queuing Telemetry Transport (MQTT). Na programação do dispositivo embarcado Arduino é utilizada a linguagem de programação C++. O levantamento de informações e os estudos da aplicação atual foram realizados por meio de pesquisas literárias sobre desenvolvimento de skills e uso de serviços da Amazon Web Services. Como forma de avaliação dos objetivos e de avaliar a experiência dos usuários foi utilizado um questionário. Ao todo a pesquisa contou com 21 participantes, que a partir de suas respostas foi possível identificar que os objetivos do trabalho foram alcançados e que a *skill* foi recebida de forma positiva por todos. Além disso foram apontados pontos de melhoria para um projeto futuro. Como resultado, obteve-se a releitura de um jogo clássico utilizando tecnologias atuais trazendo um diferencial para o jogo e maior praticidade para o usuário.

Palavras-chave: Amazon Alexa. Arduino Ameba. Skill. Internet das coisas. IOT.

ABSTRACT

This project presents the requirements gathering, specification, development, and operability of a skill for the Alexa virtual assistant. The main objective of this work is to develop a skill using the Amazon Web Services platform, in which it will be a rereading of the classic game Genius. One of the specific objectives is to use the virtual assistant feature to recognize and synthesize voice to perform all interaction between the user and the game. The interaction part of the skill was developed using the Alexa developer console, the skill logic was developed on the Amazon Web Service Lambda platform using the Python programming language. The skill communicates with the Arduino ESP32 microcontroller using the Message Queuing Telemetry Transport (MQTT) communication protocol. The C++ programming language is used in the programming of the Arduino embedded device. The survey of information and studies of the current application were carried out through literary research on skills development use of Amazon Web Services. A questionnaire was used to evaluate the objectives and evaluate the users' experience. In all, the survey included 21 participants that from their answers it was possible to identify that the objectives of the work were achieved, and that the skill was received positively by all. In addition, improvement points were pointed out for a future project. As a result, we obtained the rereading of a classic game using current technologies bringing a differential to the game and greater practicality for the user.

Keywords: Amazon Alexa. Arduino Ameba. Skill. Internet of Things. IoT.

LISTA DE FIGURAS

Figura 1 - Imagem do processamento do comando de voz na Alexa	18
Figura 2 - Jogo Genius.....	21
Figura 3 - ESP32.....	22
Figura 4 - Protótipo do jogo.....	23
Figura 5 - Diagrama de Atividades.....	25
Figura 6 - Maquete.....	26
Figura 7 - Circuito	26
Figura 8 - Diagrama de Caso de Uso.....	29
Figura 9 - Diagrama de atividades.....	31
Figura 10 - Diagrama esquemático	31
Figura 11- Carcaça do jogo da memória da empresa Unik Toys	33
Figura 12 - Jogo montado com ESP32 embutido.....	33
Figura 13 - Cadastrar slot	36
Figura 14 - Criar <i>Invocation Name</i>	45
Figura 15 - Adicionar novo <i>Intent</i>	45
Figura 16 - Adicionar nova <i>Utterance</i>	46
Figura 17 - Adicionar novo <i>slot</i>	46
Figura 18 - Configurar <i>slot type</i>	47
Figura 19 - <i>Built-In Intents</i>	47
Figura 20 – a) iniciar a <i>skill</i> – b) iniciar o jogo	51
Figura 21 – a) receber a questão a ser respondida – b) responder quantidade de cores	51
Figura 22 – a) resposta errada – b) responder à questão de sequência de cores.....	52
Figura 23 – a) resposta errada – b) informações ou regras do jogo.....	52
Figura 24 – a) registrar ou excluir pontuação – b) nomes não considerados	53
Figura 25 - Lista das pontuações registradas.....	53
Figura 26 – a) iniciar a <i>skill</i> (en-us) – b) iniciar o jogo (en-us).....	54
Figura 27 – a) questão a ser respondida (en-us) – b) responder sequência de cores (en-us)....	54
Figura 28 – a) resposta errada (en-us) – b) informações ou regras do jogo (en-us)	55
Figura 29 - Responder à questão da quantidade de cores (en-us) – b) resposta errada (en-us)	55
Figura 30 – a) registrar ou exclui pontuação (en-us) – b) pontuações registradas (en-us).....	56
Figura 31 - Primeira seção questionário	57

Figura 32 - Segunda seção questionário	58
Figura 33 - Segunda seção questionário (parte 2)	59
Figura 34 - Segunda seção questionário (parte 3)	59
Figura 35 - Terceira seção questionário	60
Figura 36 - Terceira seção questionário (parte 2)	61
Figura 37 - Terceira seção questionário (parte 3)	62
Figura 38 - Quarta seção questionário	62
Figura 39 - Quarta seção questionário (parte 2)	63
Figura 40 - Quarta seção questionário (parte 3)	64
Figura 41 - Quinta seção questionário	64
Figura 42 - Quinta seção questionário (parte 2)	65
Figura 43 - Quinta seção questionário (parte 3)	66
Figura 44 - Questionário - resultado escolaridade.....	67
Figura 45 - Questionário - resultado se conhece uma assistente virtual.....	67
Figura 46 - Questionário - resultado qual assistente virtual utiliza	68
Figura 47 - Questionário - qual frequência utiliza uma assistente virtual	68
Figura 48 - Questionário - resultado para o quê você utiliza uma assistente virtual	68
Figura 49 - Questionário - resultado comando abrir <i>skill</i>	69
Figura 50 - Questionário - resultado comando informações do jogo	69
Figura 51 - Questionário - resultado comando regras do jogo	69
Figura 52 - Questionário - resultado comando iniciar jogo	70
Figura 53 - Questionário - resultado comando receber questão	70
Figura 54 - Questionário - resultado comando responder questão	70
Figura 55 - Questionário - resultado comando registrar pontuação	71
Figura 56 - Questionário - resultado comando falar lista de pontuação.....	71
Figura 57 - Questionário - resultado comando abrir <i>skill</i> (en-us).....	71
Figura 58 - Questionário - resultado comando informações do jogo (en-us)	72
Figura 59 - Questionário - resultado comando regras do jogo (en-us)	72
Figura 60 - Questionário - resultado comando iniciar jogo (en-us).....	72
Figura 61 - Questionário - resultado comando receber questão (en-us).....	73
Figura 62 - Questionário - resultado comando responder questão (en-us).....	73
Figura 63 - Questionário - resultado comando registrar pontuação (en-us)	73
Figura 64 - Questionário - resultado comando falar lista de pontuação (en-us).....	74
Figura 65 - Questionário - resultado conclusão dos objetivos.....	74

Figura 66 - Questionário - resultado conclusão sem auxílio externo	74
Figura 67 - Questionário - resultado usabilidade para pessoas com necessidades especiais	75
Figura 68 - Questionário - resultado classificação da usabilidade	75
Figura 69 - Questionário - resultado se é um jogo simples	75
Figura 70 - Questionário - resultado modos de jogo	76
Figura 71 - Questionário - resultado críticas, comentários ou sugestões	76
Figura 72 - Questionário - resultado erro ou melhorias	77
Figura 73 - Questionário - resultado erros ou melhorias (continuação)	77

LISTA DE QUADROS

Quadro 1 - Regras de negócio da <i>skill</i>	28
Quadro 2 - Requisitos Funcionais da <i>skill</i>	29
Quadro 3 - Requisitos Não funcionais da <i>skill</i>	29
Quadro 4 - Matriz de rastreabilidade dos RF e sua relação com os Casos de Uso.....	30
Quadro 5 – Descrição das despesas	33
Quadro 6 - Código Arduino: exemplo adaptado.....	34
Quadro 7 - Código Arduino: conexão	34
Quadro 8 - Código Arduino: mensagens	35
Quadro 9 - Comandos <i>skill</i> pt-br	37
Quadro 10 - Comandos <i>skill</i> en-us.....	37
Quadro 11 - Questionário	39
Quadro 12 - Respostas questionário	39
Quadro 13 - Comparativo trabalhos correlatos.....	40
Quadro 14 - Descrição do UC01	48
Quadro 15 - Descrição UC02	48
Quadro 16 - Descrição do UC03	48
Quadro 17 - Descrição do UC04	49
Quadro 18 - Descrição do UC05	49
Quadro 19 - Descrição do UC06	49
Quadro 20 - Descrição do UC07	49
Quadro 21 - Descrição do UC08	50
Quadro 22 - Descrição do UC09	50

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

AWS – Amazon Web Services

HTTPS – HyperText Transfer Protocol Secure

IFTT – If This Then That

IoT – Internet of Things

JSON – JavaScript Object Notation

LED – Light-Emitting Diode

MQTT – Message Queue Telemetry Transport

PIC – Programmable Interrupt Controller


UML – Unified Modeling Language

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVOS.....	16
1.2 ESTRUTURA.....	16
2 FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 AMAZON ALEXA	18
2.2 AMAZON WEB SERVICES.....	18
2.2.1 Conceitos de desenvolvimento da <i>skill</i>	19
2.2.2 AWS Lambda.....	20
2.2.3 DynamoDB.....	20
2.2.4 CloudWatch.....	20
2.2.5 AWS IoT Core	20
2.3 JOGO GENIUS	21
2.4 ESP32	21
2.4.1 Controlador Inteligente	22
2.5 TRABALHOS CORRELATOS.....	22
2.5.1 Jogo da memória embarcado multinível	23
2.5.2 Protótipo de automação residencial utilizando uma assistente de voz.....	24
2.5.3 Automação residencial de baixo custo	25
3 DESENVOLVIMENTO.....	28
3.1 LEVANTAMENTO DE REQUISITOS.....	28
3.2 ESPECIFICAÇÃO	28
3.2.1 Requisitos	28
3.2.2 Diagrama de Casos de Uso.....	29
3.2.3 Matriz de rastreabilidade dos RF e sua relação com os Casos de Uso.....	30
3.2.4 Diagrama de atividades.....	30
3.2.5 Diagrama esquemático	31
3.3 IMPLEMENTAÇÃO.....	32
3.3.1 Técnicas e ferramentas utilizadas.....	32
3.3.2 Operacionalidade da implementação	37
3.4 RESULTADOS E DISCUSSÕES.....	37
3.4.1 Avaliação de usabilidade por meio de questionário eletrônico	38

3.4.2 Comparação entre os trabalhos correlatos e o trabalho desenvolvido	39
4 CONCLUSÕES.....	41
4.1 EXTENSÕES	41
REFERÊNCIAS.....	43
APÊNDICE A – COMO CADASTRAR ITENS ESSENCIAIS PARA O FUNCIONAMENTO DA <i>SKILL</i>.....	45
APÊNDICE B – DESCRIÇÃO DOS CASOS DE USO	48
APÊNDICE C – EXECUÇÃO DOS COMANDOS NO CONSOLE DO DESENVOLVEDOR ALEXA	51
APÊNDICE D – ROTEIRO DE AVALIAÇÃO DA <i>SKILL</i>	57
APÊNDICE E – PERGUNTAS E RESPOSTAS OBTIDAS NA AVALIAÇÃO UTILIZANDO QUESTIONÁRIO	67


1 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 (2018), 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, Wencelewski e Paixão (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.

Se pode 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 (LUIZ, 2019).



A tecnologia de assistentes virtuais não se limita apenas a smartphones, já sendo possível ter acesso a outros dispositivos eletrônicos que possuem a tecnologia de 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 (disponível no Amazon Echo) 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, desenvolveu-se um estudo sobre a assistente de voz Alexa criando uma *skill* que irá reproduzir algumas das funções do jogo Genius. Para isto se definiu uma arquitetura que é composta pela Alexa, controlador inteligente (ESP32) e um conjunto de lâmpadas Light Emitting Diodes (LED's). Isso permitiu definir um meio de interação alternativa referente ao jogo original aonde a ação do usuário é clicar nas cores corretas com as mãos. Consequentemente também é feita uma modernização do jogo clássico que irá abranger toda e qualquer pessoa que tenha interesse em jogar e testar sua capacidade de memorização.

1.1 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.

Os objetivos específicos são:

- a) avaliar o uso do recurso da assistente virtual para reconhecer e sintetizar voz, permitindo toda interação do usuário ser exclusivamente por voz, em português brasileiro e inglês estadunidense; 
- b) verificar se ao receber uma sequência de palavras por voz seria possível validar se era o valor esperado ou não; 
- c) avaliar a integração da *skill* com outros serviços da Amazon: DynamoDB; IoT Core; AWS Lambda; e CloudWatch;
-   d) criar um tutorial no idioma português brasileiro de criação e *skills*.

1.2 ESTRUTURA

O trabalho está organizado em quatro capítulos. O primeiro capítulo apresenta a introdução do trabalho desenvolvido, os objetivos a serem alcançados e sua estrutura. No segundo capítulo são abordadas as fundamentações teóricas para a construção deste trabalho, tendo tópicos como o que é Amazon Alexa, Amazon Web Services (AWS), serviços disponibilizados na AWS, conceitos necessários no desenvolvimento de uma *skill*, o jogo que inspirou o trabalho e componente eletrônico necessário para o desenvolvimento do projeto, e os trabalhos correlatos. O terceiro capítulo descreve o desenvolvimento do trabalho com os

requisitos levantados; a especificação do trabalho por meio de diagramas; as técnicas e ferramentas utilizadas; síntese do funcionamento geral da *skill* e a operacionalidade da implementação. Neste capítulo ainda é exposto os resultados obtidos por meio de avaliações dos usuários e as dificuldades encontradas no desenvolvimento. Por fim, o quarto e último capítulo finaliza com as principais conclusões sobre o trabalho desenvolvido e com possíveis extensões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais assuntos que estão relacionados com o trabalho proposto. A seção 2.1 aborda uma visão geral sobre a assistente virtual Alexa. A seção 2.2 trata de apresentar o que é a Amazon Web Services, principais conceitos ao se desenvolver uma *skill* e os serviços utilizados. Já na seção 2.3 explica o jogo Genius ~~e de onde surgiu a ideia de usar este jogo para o desenvolvimento deste trabalho~~. A seção 2.4 apresenta o que é o ESP32 utilizado como um controlador inteligente. E, pôr fim na seção 2.5, são apresentados os trabalhos correlatos.

2.1 AMAZON ALEXA

A Alexa é o serviço de voz baseado em nuvem da Amazon disponível em dezenas de milhões de dispositivos da própria Amazon e de fabricantes de dispositivos de terceiros (AMAZON ALEXA, 2021). Com a Alexa 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 outras diversas tarefas, 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 1 apresenta a sequência de passos do funcionamento ao realizar um comando por voz de uma *skill*.



Fonte: Realtek (2012).

2.2 AMAZON WEB SERVICES

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

De acordo com 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.

2.2.1 Conceitos de desenvolvimento da *skill*

Nesta subseção será listado os principais conceitos ao se criar uma *skill* para a Alexa, e ao entender esses conceitos o desenvolvedor já estará apto para criar uma *skill* personalizada. No Apêndice A se encontra uma descrição como criar cada item do que será explicado abaixo.

O primeiro conceito seria o `Invocation Name`, que é a palavra ou frase usada para acionar sua habilidade. De certa forma, é o equivalente da voz a um ícone de aplicativo. o `Invocation Name` geralmente corresponde ao nome da sua habilidade, mas dadas as regras sobre a escolha de um `Invocation Name`, poderá ser um pouco diferente (ALEXA DEVELOPERS, 2020).

O próximo conceito `Intent` é o que um usuário está tentando realizar. Dentro do código a `Intent` é o método a ser executado. A `Intent` não se relaciona com as palavras específicas que um usuário diz, mas com o objetivo de alto nível que ele almeja (ALEXA DEVELOPERS, 2020).

Já a `Utterance` são as frases específicas que os usuários usarão ao fazer uma solicitação após abertura da *skill* e que por consequência executará uma determinada `Intent`. 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).

Ainda, 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). A Amazon fornece vários tipos de `slots` integrados, como datas, números, durações, hora entre outros. Mas os desenvolvedores podem criar `slots` personalizados para variáveis específicas para sua *skill* (conceito é chamado de `slot type`) (ALEXA DEVELOPERS, 2020).


E, por fim, `Built-in Intents` 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` e `AMAZON.RepeatIntent` (ALEXA DEVELOPERS, 2020).

2.2.2 AWS Lambda

Um dos serviços disponibilizados pela Amazon é o AWS Lambda, que segundo a 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 este código seja acionado automaticamente por meio de outros produtos da AWS ou chamá-lo diretamente usando qualquer aplicação móvel ou da Web.

2.2.3 DynamoDB



Para a persistência de dados após o uso da *skill* é utilizado o serviço DynamoDB. 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 (AMAZON DYNAMODB, 2022).

2.2.4 CloudWatch

Para auxílio do desenvolvimento da *skill* é vinculado o serviço CloudWatch, que com este serviço é possível gerar *logs* e acompanhar em tempo real a execução da sua *skill*. Entre outras características são possíveis citar: criação de alarmes, coletar e monitorar métricas de suas aplicações, e criação de painéis com as métricas escolhidas (AMAZON CLOUDWATCH, 2022).

2.2.5 AWS IoT Core

O AWS IoT Core é um gerenciamento na nuvem que permite a conexão com outros aparelhos e serviços. Segundo Amazon Whitepaper (2022), o AWS IoT Core pode oferecer suporte a bilhões de dispositivos e trilhões de mensagens. E ainda pode processar e rotear essas mensagens para *endpoints* da AWS e outros dispositivos de maneira confiável e segura utilizando os protocolos MQTT ou HTTPS.

2.3 JOGO GENIUS

O jogo Genius era um brinquedo muito popular na década de 80 no Brasil, e que é ainda comercializado até hoje. Lançado pela Brinquedos Estrela SA o jogo Genius buscava estimular a memorização de cores e sons. O jogo Genius tem um formato simples e oval, com botões coloridos que emitiam sons harmônicos e se iluminam em sequência (Figura 2).

Figura 2 - Jogo Genius



Fonte: VejaSP (2017).

O objetivo do jogo Genius é reproduzir a sequência de cores apresentada previamente sem errar. O jogo conta 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 da escolha do jogo Genius para o desenvolvimento deste trabalho se deve ao fato que este jogo estimula a memória. 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: desenvolver habilidades de concentração, autonomia e confiança.

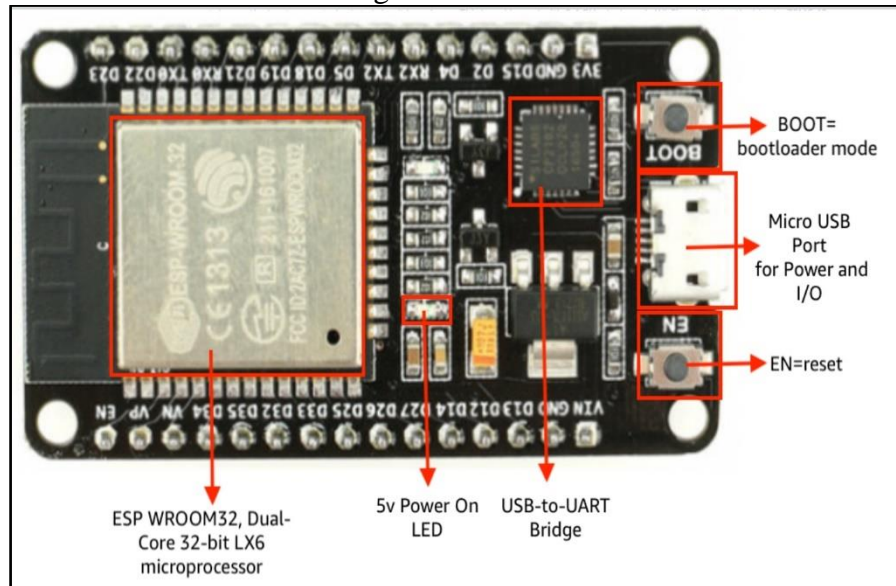
2.4 ESP32

Desenvolvido pela empresa Espressif, o ESP32 é um dispositivo IoT 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 este dispositivo possa ser programado de forma independente, sem a necessidade de outras placas microcontroladores 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).

De acordo com 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* e gratuita. Neste projeto o ESP32 (Figura 3) foi utilizado como uma controladora inteligente.

Figura 3 - ESP32



Fonte: Amazon Workshop (2022).

2.4.1 Controlador Inteligente



Um controlador inteligente é um interruptor tradicional que possui boa capacidade de processamento e fácil integração com demais computadores. O intuito do controlador é gerenciar os aparelhos a ele conectado, sendo usualmente lâmpadas. E com isso, o controlador pode além de gerenciar o acionamento das lâmpadas, também permite controlar a intensidade da luz. Permite ainda criar uma política de horários para acender ou apagar as lâmpadas 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 se vincular a Alexa neste componente é possível realizar os comandos por voz para tentar tornar a experiência mais dinâmica, simples e conveniente. Usualmente, é feito o comando direto por voz para ligar e desligar as lâmpadas, mas neste caso será executado o jogo e ele irá definir qual será a ação da controladora.

2.5 TRABALHOS CORRELATOS

Nesta seção são descritos três trabalhos correlatos que apresentam propostas semelhantes com o trabalho desenvolvido. Na subseção 2.5.1 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. Na subseção 2.5.2 está descrito o protótipo desenvolvido por Dallarosa Neto (2018), que tem como principal característica o uso da Alexa como intermediária do Arduíno. Por fim, na subseção 2.5.3 consta uma automação residencial feita por Moura Junior *et al.* (2020) que utiliza a assistente virtual do Google.

2.5.1 Jogo da memória embarcado multinível

No 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) 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 4), sendo a parte lógica com a linguagem de programação C. 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 (2008), 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 4 - Protótipo do jogo



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 acrescentado um novo elemento a fim de ter um sistema de fases e aumento da dificuldade ao jogador. Para os testes do protótipo foi utilizado ferramentas específicas para validar todos os periféricos, como o display, *led*, botões e *buzzer*. E após estes testes de

montagem o protótipo foi disponibilizado para um grupo de 20 pessoas jogarem uma partida em cada nível.

Segundo Zuffo (2008) o projeto se mostrou 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. Por fim, 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 (ZUFFO, 2008).

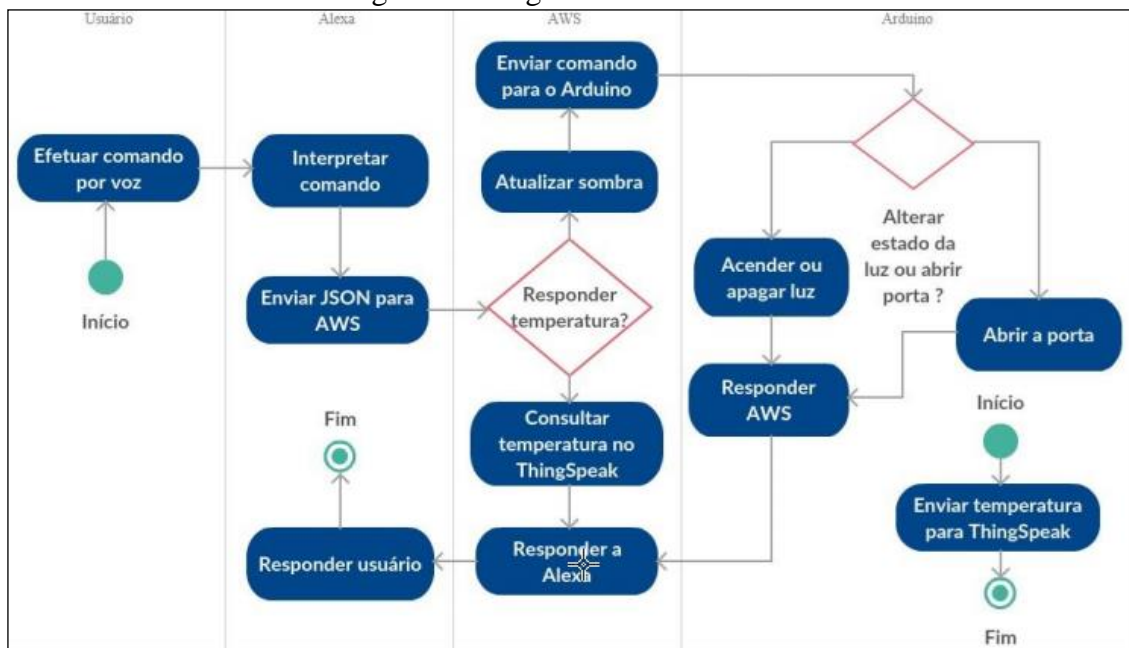
2.5.2 Protótipo de automação residencial utilizando uma assistente de voz

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 5 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: usuário que requisita o comando; o aplicativo Alexa que interpreta e envia o comando aonde posteriormente retorna uma mensagem de erro ou sucesso ao usuário; AWS que processa o comando e retorna à Alexa ou envia a ação ao Arduino; e por fim, o Arduino que executa a ação solicitada, assim retornando à AWS resultado de sucesso ou não.

Para usar o protótipo deve-se iniciar com um 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 (DALLAROSA NETO, 2018).

Figura 5 - Diagrama de Atividades



Fonte: Dallarosa Neto (2018).

De acordo com Dallarosa Neto (2018), um dos maiores desafios a realizar o desenvolvimento deste protótipo foi: 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 do protótipo, 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.

2.5.3 Automação residencial de baixo custo

O trabalho de Moura Junior *et al.* (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 (Figura 6), segundo Moura Junior *et al.* (2020), foi inspirado em uma casa moderna tipo sobrado já projetada para projetos dessa natureza. Pois esta maquete 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.

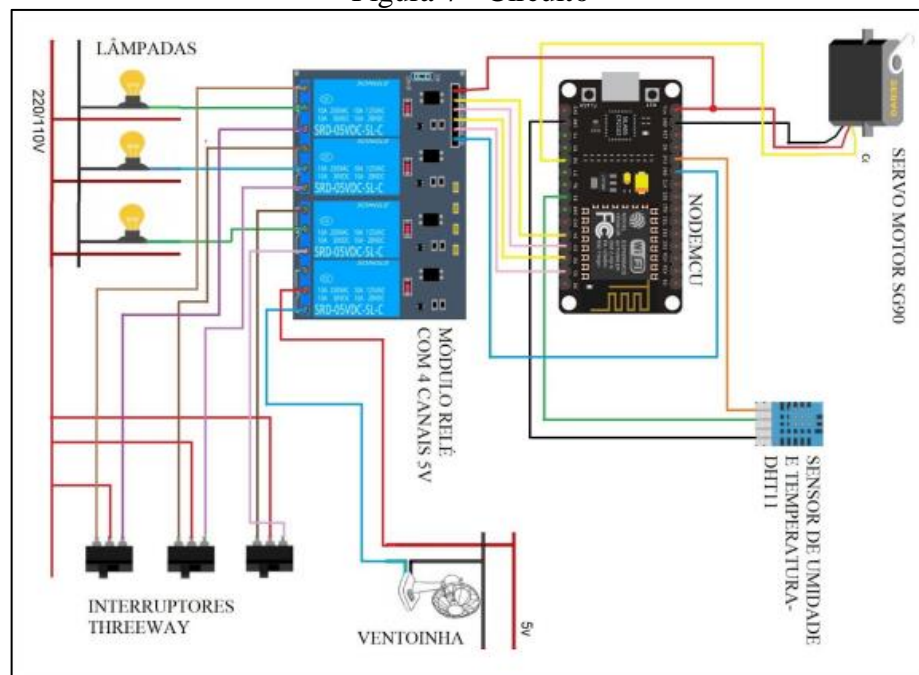
Figura 6 - Maquete



Fonte: Moura Junior *et al.* (2020).

A montagem do circuito (Figura 7) 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.

Figura 7 - Circuito



Fonte: Moura Junior *et al.* (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 Moura Junior *et al.* (2020), com o IFTTT é possível criar instruções condicionais chamadas *applets* e unir diferentes aplicativos da web como o


Google e o Facebook em um simples comando. Comando este sendo utilizado como 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 usando o 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. Já 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 (Moura Junior *et al.* 2020).

3 DESENVOLVIMENTO

No presente capítulo são descritas as etapas para o desenvolvimento deste projeto. Na seção 3.1 contém o levantamento dos requisitos, na seção 3.2 contém a especificação detalhada da *skill*, já na seção 3.3 tem detalhes da implementação abordando ambos os códigos fontes, a preparação do projeto físico e a operacionalidade da *skill*, e na seção 3.4 é feita uma análise dos resultados obtidos da *skill*.


3.1 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, e salvar os dados no banco de dados.

3.2 ESPECIFICAÇÃO

Nesta seção são apresentadas as especificações técnicas da aplicação e os diagramas deste trabalho. Para tal, os requisitos são apresentados na subseção 3.2.1, contendo as Regras de Negócio (RN), os Requisitos Funcionais (RF) e os Requisitos Não Funcionais (RNF). Na subseção 3.2.2 é apresentado o Diagrama de Caso de Uso (DCU) criado na ferramenta Draw.io. Na subseção 3.2.3 é exposta a matriz de rastreabilidade entre os Casos de Uso (CU) e sua relação com os RF. Na subseção 3.2.4 é exibida o diagrama de atividades também criado na ferramenta Draw.io. Por fim, é demonstrado o diagrama esquemático na subseção 3.2.5 feito na ferramenta Circuito.io.

3.2.1 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.

Fonte: elaborado pelo autor.

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 <i>skill</i> .
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.

Fonte: elaborado pelo autor.

Quadro 3 - Requisitos Não funcionais da *skill*

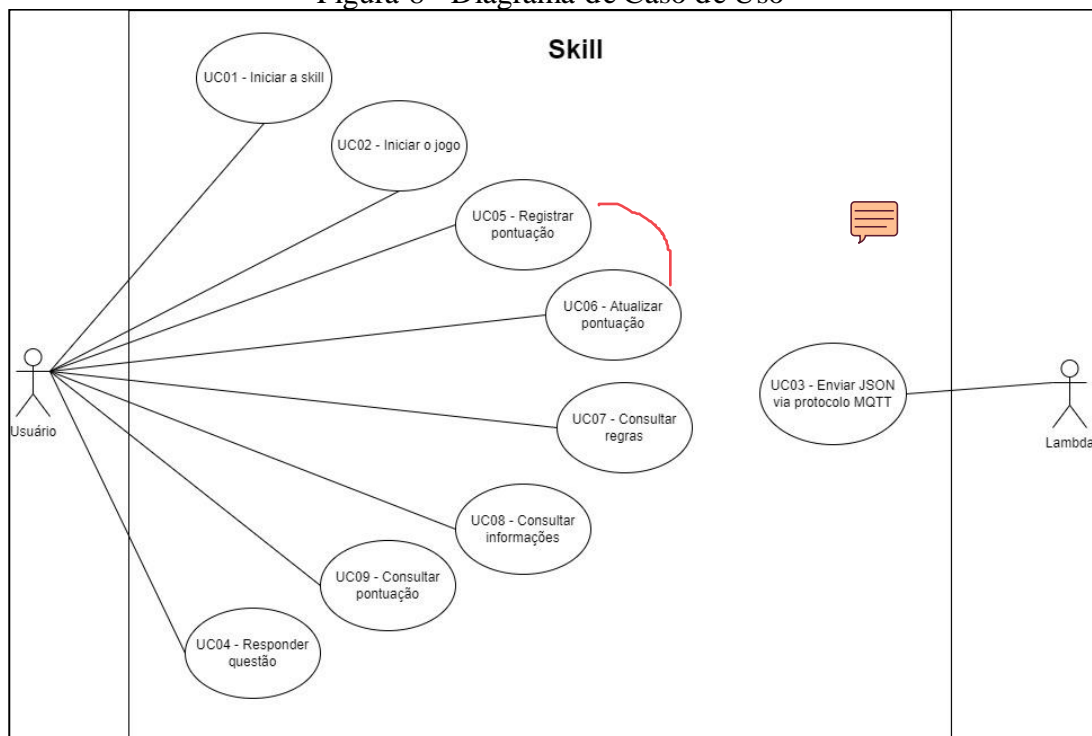
RNF	O sistema deverá:
RNF01	Atualizar a maior pontuação do usuário no banco.
RNF02	Ser desenvolvido em Python e na própria plataforma disponibilizada pela Amazon.
RNF03	Utilizar a plataforma da Amazon Web Services (AWS).
RNF04	Utilizar o serviço CloudWatch da AWS.
RNF05	Utilizar o serviço DynamoDB da 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.

Fonte: elaborado pelo autor.

3.2.2 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 B. A Figura 8 apresenta o diagrama de casos de uso das funcionalidades do protótipo.

Figura 8 - Diagrama de Caso de Uso



Fonte: elaborado pelo autor.

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.

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

Esta subseção apresenta o Quadro 4 que exibe a matriz de rastreabilidade dos RF com os UCs referente a *skill* desenvolvida.

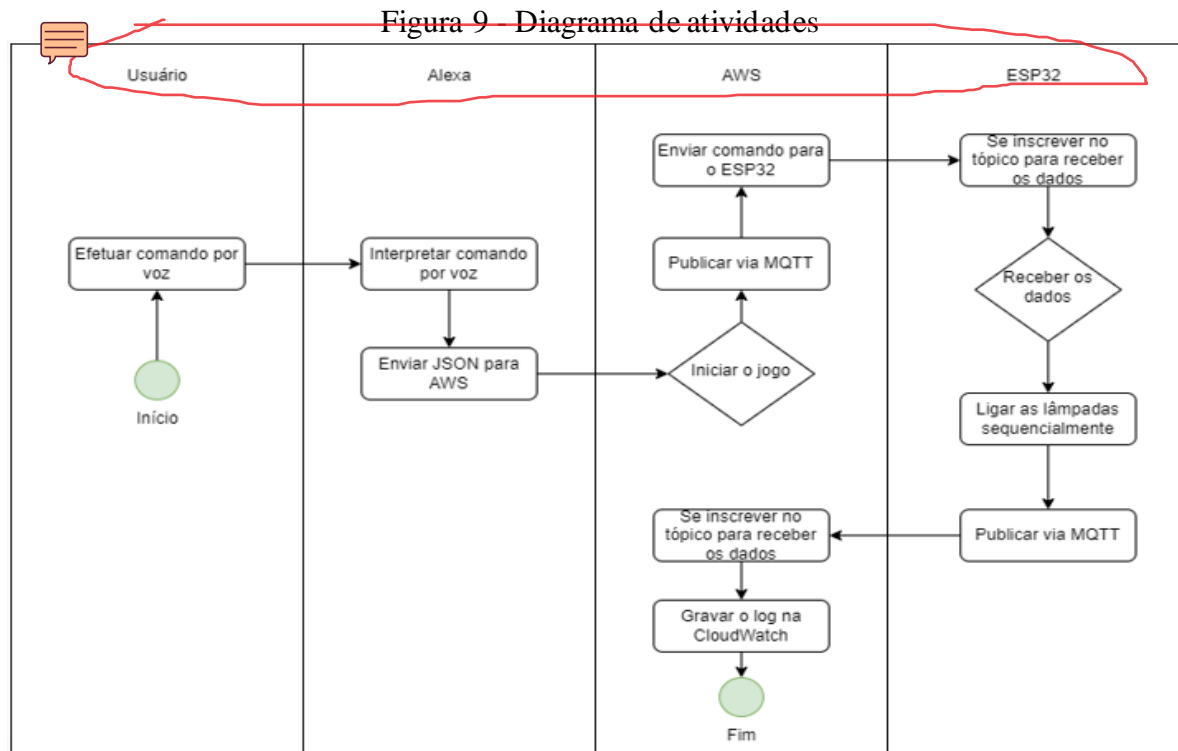
Quadro 4 - Matriz de rastreabilidade dos RF e sua relação com os Casos de Uso

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

Fonte: elaborado pelo autor.

3.2.4 Diagrama de atividades

O diagrama de atividades da Figura 9 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*.

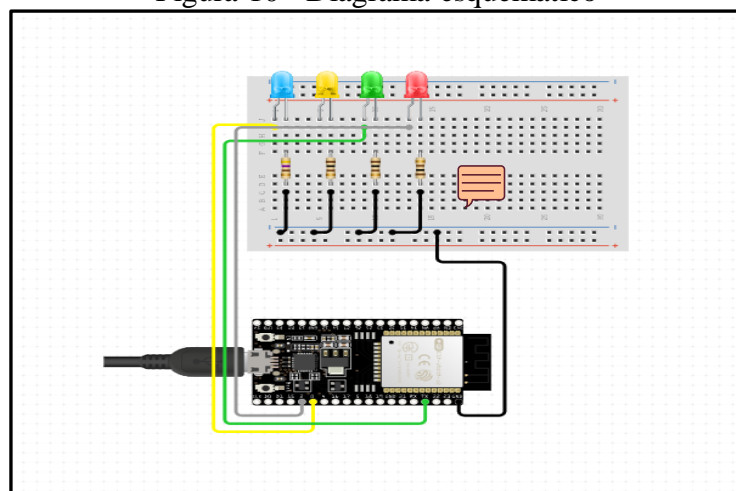


Fonte: elaborado pelo autor.

3.2.5 Diagrama esquemático

O diagrama esquemático do circuito do protótipo foi todo desenhado utilizando a ferramenta Circuito.io (Figura 10). 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 fonte externa. Também é feita a conexão “terra” entre a *protoboard* e o Arduino.

Figura 10 - Diagrama esquemático



Fonte: elaborado pelo autor.

3.3 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, conteúdo também a codificação da *skill*; e, por fim, a subseção 3.3.2 apresenta a operacionalidade da implementação.

3.3.1 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 (item 2.2.1).

Após esta primeira etapa, foi realizada a análise dos requisitos para a construção da *skill*, estando presentes no item 3.2.1. 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.

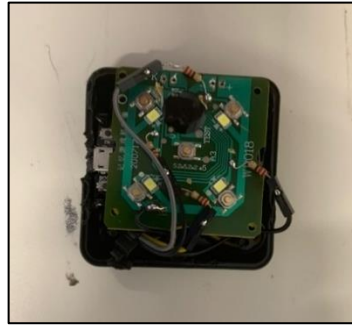
Como material complementar contendo tutorial, códigos fontes, JSON com as frases da *skill* e exemplos de testes utilizando os recursos da Amazon foi criado um repositório que pode ser encontrado em Rodrigues (2022).

3.3.1.1 Preparação do circuito eletrônico para a maquete

Para a criação da carcaça do trabalho foi utilizado uma parte do jogo da memória (da empresa Unik Toys) e a placa de circuitos, e ainda foi retirado outros componentes desnecessários. O projeto físico conta também com 4 LEDs para serem utilizados para representar as sequências de cores.

Na Figura 10 se pode visualizar a representação do circuito antes da montagem com a carcaça, na Figura 11 a carcaça do jogo da memória da empresa Unik Toys, e por fim, na Figura 12 o projeto físico montado. Em resumo, foi utilizado na montagem do projeto os seguintes componentes apresentados no Quadro 5.

Figura 11- Carcaça do jogo da memória da empresa Unik Toys



Fonte: elaborado pelo autor.

Figura 12 - Jogo montado com ESP32 embutido



Fonte: elaborado pelo autor.

Quadro 5 – Descrição das despesas

Descrição	Qtd.	Preço	
ESP-WROOM-32 DEVKIT V11 WIFI e Bluetooth – NODEMCU	1	R\$ 84,43	R\$ 84,43
Resistências de 220ohms	4	R\$ 0,10	R\$ 0,40
Cabo Micro USB	1	R\$ 19,59	R\$ 19,59
Jumper fêmea x fêmea (kit 10 unidades de 20 cm)	1	R\$ 3,90	R\$ 3,90
Jogo da memória Unik Toys	1	R\$ 53,24	R\$ 53,24
Total		R\$ 161,56	

Fonte: elaborado pelo autor.

3.3.1.2 Programando o Arduino Ameba

Os métodos essenciais que deve existir no código do Arduino e consequentemente os primeiros a serem executados seria `setup` e `loop`. Em Rodrigues (2022) contém mais informações sobre os métodos, configuração e funcionamento do código fonte utilizado, além do descrito a seguir.

O `setup` (133) é executado sempre que o Arduino é iniciado. O `loop` (139) é um laço que é constantemente executado após ele mesmo terminar a execução do `setup`. O Quadro 6 mostra parte do código do método `setup` adaptado de um exemplo gerado dentro da Arduino IDE. Neste código é executado dois métodos para tentar se conectar ao Wi-Fi, e posteriormente a Amazon e escolher os conectores do ESP32 que serão utilizados.

Quadro 6 - Código Arduino: exemplo adaptado

```

118     }
119     Serial.println("The Pieces separated by strtok()");
120     for (int n = 0; n < index; n++)
121     {
122         Serial.print(n);
123         Serial.print(" ");
124         Serial.println(strings[n]);
125         digitalWrite(strings[n], HIGH);
126         delay(delayBulb);
127         digitalWrite(strings[n], LOW);
128         delay(100);
129     }
130     publishReturnMessage();
131 }
132
133 void setup() {
134     Serial.begin(9600);
135     connectAWS();
136     setupBulb();
137 }
138
139 void loop() {
140     if (!client.connected()) {reconnect();}
141     client.loop();
142     delay(1000);
143 }
144

```

Fonte: elaborado pelo autor.

Já no Quadro 7 consta o método `connectAWS` (14) que utiliza as bibliotecas `WiFi` para primeiramente conectar-se à rede. Ainda no Quadro 7 tem método `WiFiClientSecure` para utilizar os certificados (33, 34 e 35) gerados na Amazon e conseguir se conectar na `Thing` e posteriormente se inscrever no tópico.

Quadro 7 - Código Arduino: conexão

```

14 void connectAWS()
15 {
16     WiFi.mode(WIFI_STA);
17     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
18
19     Serial.println("Connecting to Wi-Fi");
20
21     while (WiFi.status() != WL_CONNECTED) {
22         delay(500);
23         Serial.print(".wifi");
24     }
25
26     if (WiFi.waitForConnectResult() == WL_CONNECTED) {
27         Serial.print(", WiFi connected, IP address: "); Serial.println(WiFi.localIP());
28     } else {
29         Serial.println(", failed to connect to WiFi");
30     }
31
32     // Configure WiFiClientSecure to use the AWS IoT device credentials
33     net.setCACert(AWS_CERT_CA);
34     net.setCertificate(AWS_CERT_CRT);
35     net.setPrivateKey(AWS_CERT_PRIVATE);
36     // Connect to the MQTT broker on the AWS endpoint we defined earlier
37     client.begin(AWS_IOT_ENDPOINT, 8883, net);
38     // Create a message handler
39     client.onMessage(messageHandler);
40
41     Serial.print("Connecting to AWS IOT");
42
43     while (!client.connect(THINGNAME)) {
44         Serial.print(".");
45         delay(100);
46     }
47
48     if (!client.connected()) {
49         Serial.println("AWS IoT Timeout!");
50         return;
51     }
52     // Subscribe to a topic
53     client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC);
54     Serial.println("AWS IoT Connected!");
55 }

```

Fonte: elaborado pelo autor.

O Quadro 8 apresenta a rotina `messageHandler` (99) que irá receber o tópico e o arquivo JSON que foi enviado do servidor da AWS para o Arduino. Está rotina irá basicamente

receber o JSON, transformar 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 (125)` utilizando como segundo parâmetro o valor `HIGH` e para apagar o valor `LOW`, sendo que esses valores dependem do circuito eletrônico.

Por fim, é executado o método `publishReturnMessage (130)` que irá enviar um novo JSON ao servidor da Amazon para geração de históricos. 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.

Quadro 8 - Código Arduino: mensagens

```

89 void publishReturnMessage ()
90 {
91     StaticJsonDocument<200> doc;
92     doc["finish"] = "OK_ALEXA";
93     //doc["language"] = "en-US";
94     char jsonBuffer[512];
95     serializeJson(doc, jsonBuffer);
96     client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer);
97 }
98
99 void messageHandler(String &topic, String &payload) {
100     Serial.println("Log incoming: " + topic + " - " + payload);
101
102     StaticJsonDocument<200> doc;
103     deserializeJson(doc, payload);
104     String sequence = doc["sequence"];
105     char str_array[sequence.length()];
106     sequence.toCharArray(str_array, sequence.length());
107     const String location = doc["location"];
108     const int delayBulb = doc["delay"];
109     byte index = 0;
110     uint8_t strings[128]; // an array of pointers to the pieces of the above array after strtok()
111     char *ptr = NULL;
112     ptr = strtok(str_array, ","); // delimiter
113     while (ptr != NULL)
114     {
115         strings[index] = (uint8_t)atoi(ptr);
116         index++;
117         ptr = strtok(NULL, ",");
118     }
119     Serial.println("The Pieces separated by strtok()");
120     for (int n = 0; n < index; n++)
121     {
122         Serial.print(n);
123         Serial.print(" ");
124         Serial.println(strings[n]);
125         digitalWrite(strings[n], HIGH);
126         delay(delayBulb);
127         digitalWrite(strings[n], LOW);
128         delay(100);
129     }
130     publishReturnMessage ();
131 }
132

```

Fonte: elaborado pelo autor.

3.3.1.3 Estrutura da Amazon Web Services

Para o desenvolvimento deste projeto foi necessário entender e utilizar os conceitos expostos na seção 2.2.1. Se utilizou como base o curso “Desenvolvimento Alexa - 15 Skills Programadas em Python” disponibilizado na plataforma Udemy e a lista de vídeos “Zero to Hero (em tradução livre ‘Do Zero ao Herói’)” do canal oficial dos desenvolvedores da Alexa (ALEXA DEVELOPERS, 2020).

No entanto, por essas ferramentas (Console do desenvolvedor Alexa, AWS Lambda, AWS IoT, DynamoDB) estarem em constante 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 13. 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*, e como exemplo a Figura 13 mostra um *slot num* que aceita apenas números.

Figura 13 - Cadastrar slot

The screenshot shows the 'ScoreNumberIntent' configuration page in the Amazon Alexa Developer Console. The left sidebar lists various intents, with 'ScoreNumberIntent' selected. The main area shows 'Sample Utterances (5)' with examples like 'it shown (num) times' and 'I think the answer is (num)'. Below this is the 'Dialog Delegation Strategy' set to 'fallback to skill setting'. At the bottom, the 'Intent Slots (1)' table is visible:

ORDER	NAME	SLOT TYPE	MULTI-VALUE	ACTIONS
1	num	AMAZON.NUMBER	<input type="checkbox"/>	Edit Dialog Delete

Fonte: elaborado pelo autor.

Finalmente, 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.

3.3.2 Operacionalidade da implementação

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

No Quadro 9 consta as principais frases em português brasileiro (pt-br) a serem ditas para realizar determinada ação, sendo que a palavra envolvida por chave é o `slot`.

Quadro 9 - Comandos *skill* pt-br

AÇÃO	COMANDO
Iniciar a <i>skill</i>	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}

Fonte: elaborado pelo autor.

Já o Quadro 10 consta com as frases em inglês estadunidense (en-us).

Quadro 10 - Comandos *skill* en-us

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

Fonte: elaborado pelo autor.

Conforme Quadro 9 e Quadro 10 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 C consta a execução dos comandos no console do desenvolvedor da Alexa e o retorno esperado de cada comando.

3.4 RESULTADOS E DISCUSSÕES

A subseção 3.4.1 apresenta a avaliação de experiência e usabilidade do uso por meio de um questionário eletrônico e os resultados alcançados, a subseção 3.4.2 aborda a comparação entre os trabalhos correlatos e o trabalho desenvolvido.

3.4.1 Avaliação de usabilidade por meio de questionário eletrônico

Para avaliar e validar se a *skill* desenvolvida para a assistente virtual Alexa atingiu os objetivos deste trabalho, foi realizada uma avaliação por questionário, tendo a experiência do usuário no uso da *skill* como base.

O questionário conta com 30 questões que devem ser respondidas de maneira qualitativa e quantitativa. Foram realizadas 21 avaliações com pessoas escolhidas de **forma aleatória**, uma vez que o trabalho desenvolvido não possui um público-alvo definido. As pessoas que responderam a avaliação são consideradas usuários não-especialistas pois não foi possível saber com antecedência se a pessoa já estava familiarizada com assistentes virtuais e o jogo Genius. As avaliações ocorreram por meio da ferramenta on-line Google Forms, e o roteiro da avaliação pode ser encontrado no Apêndice D.



O questionário é dividido em quatro partes. A primeira etapa busca traçar um perfil dos usuários participantes da avaliação, solicitando as seguintes informações: escolaridade; se conhece o que é uma assistente virtual; qual assistente virtual que utiliza; qual a frequência de uso de uma assistente virtual; e para que a pessoa utiliza a assistente virtual. As respostas deste questionário estão presentes no Apêndice E.

Em relação a escolaridade 5 participantes responderam que possuem ensino médio completo - 2º grau, representando 23,8%; 10 ensino superior incompleto, correspondendo a 47,6%; e 6 ensino superior completo, equivalendo a 28,6%.

Já em relação a questão se os participantes já conheciam o que é uma assistente virtual, 15 responderam que já conhecem e utilizam; 6 que já utilizaram, mas não conhecem muito sobre assistente virtual; e nenhuma respondeu que não conhece.

Para a questão de qual assistente virtual que utilizam, 11 responderam que utilizam a Alexa, 8 utilizam a assistente virtual do Google, 3 que utilizam a assistente virtual Siri, 2 utilizam a assistente virtual Cortana e apenas 1 participante que não utiliza nenhuma. Desse percentual que conhece apenas 4 utilizam frequentemente, conforme figuras Figura 46 e Figura 47.

Por fim, nesta seção é questionado para o quê utilizam a assistente virtual sendo que 13 pessoas responderam que utilizam para executar o aplicativo de música, 5 para ativas automações residenciais, 1 para jogar, 1 para realizar compras, 1 para escutar receitas, 1 para cadastrar lembretes, 1 para notícias, 1 para enviar responder mensagens, e 4 participantes responderam que não utilizam. Esses dados podem ser vistos na Figura 48.

A segunda etapa se refere ao tutorial de como utilizar a *skill* português brasileiro. Dentro dos comandos requisitados, apenas o de registrar a pontuação do usuário que 2 participantes não conseguiram executar por causa que a Alexa não reconhecia o nome falado. Os outros comandos todos foram executados com sucesso.

A terceira etapa seria sobre o uso da *skill* no idioma inglês estadunidense. Em média geral apenas 50% dos participantes conseguiram realizar todos os comandos sem problemas. Na questão disponibilizada para críticas, sugestões ou melhorias, consta que a Alexa não conseguiu entender o sotaque, ou seja, não seria algo que a *skill* possa controlar.

A quarta e última etapa seria sobre a avaliação do usuário referente a *skill*. Apenas 4 participantes não conseguiram realizar os comandos da *skill* sem ajuda externa, já os outros 17 conseguiram. No Quadro 11 consta as questões quantitativas da avaliação e no Quadro 12 é demonstrado os resultados obtidos com os questionamentos do Quadro 11. Para facilitar a visualização dos resultados, os números foram arredondados, pois não influenciam no resultado da avaliação.

Quadro 11 - Questionário

Perguntas da avaliação	
P1	Você acredita que pessoas com necessidades especiais (cegos e baixa visão) se beneficiariam com esta proposta de interface diferente, usando <i>skill</i> ?
P2	O usuário só poderá responder à questão após o comando. Como você classifica a usabilidade da <i>Skill</i> - Jogo da Memória?
P3	Você acha que a <i>skill</i> cumpriu o objetivo de ser um jogo simples com diferencial da assistente virtual?
P4	Você acha que faltou mais modos de jogo?

Fonte: elaborado pelo autor.

Quadro 12 - Respostas questionário


Número	0	1	2	3	4	5
P1	N/A	0%	0%	5%	43%	52%
P2	N/A	0%	0%	29%	38%	33%
P3	0%	0%	5%	10%	33%	52%
P4	24%	5%	24%	9%	29%	9%

Fonte: elaborado pelo autor.

3.4.2 Comparação entre os trabalhos correlatos e o trabalho desenvolvido

Nesta subseção consta o Quadro 13 que apresenta um comparativo entre os trabalhos correlatos e o trabalho desenvolvido, de modo que as linhas representam as características e as colunas os trabalhos relacionados.

Quadro 13 - Comparativo trabalhos correlatos



Correlatos Características	Zuffo (2008)	Dallarosa Neto (2018)	Moura Junior <i>et al.</i> (2020)	Rodrigues (2022)
Assistente Virtual	X	Alexa	Google Assistant	Alexa
Jogo	Genius	X	X	Genius
Automação residencial	X	✓	✓	X
Plataforma	C	JavaScript/C++	C++	Python/C++

Fonte: elaborado pelo autor.

Conforme apresentado no Quadro 13, percebe-se que apenas o Zuffo (2008) não utiliza uma Assistente virtual no desenvolvimento do trabalho, mas foi o único que tenta realizar a releitura do Jogo Genius. Já em Dallarosa Neto (2018) e Rodrigues (2022) utilizam o recurso da Assistente Virtual divergindo apenas de Moura Junior *et al.* (2020), sendo este item uma característica identificada de suma importância para alcançar os objetivos. Rodrigues (2022) se diferencia pelo uso da assistente virtual que em vez de utilizar como controle de Automação residencial utiliza como forma de interação para a releitura do Jogo Genius. Todos os trabalhos utilizam a linguagem de programação C/C++ para a programação do Arduino, já para a programação da lógica da assistente virtual Dallarosa Neto (2018) utiliza JavaScript, Rodrigues (2022) Python e Moura Junior *et al.* (2020) não especifica qual utilizou.

4 CONCLUSÕES

O presente trabalho de conclusão de curso apresenta o desenvolvimento de uma *skill* para a assistente virtual Alexa com o intuito de realizar a releitura do jogo Genius. O catálogo de serviços da Amazon Web Service: AWS Lambda, console do desenvolvedor Alexa, DynamoDB, IoT Core, CloudWatch são ferramentas poderosas e úteis que atenderam a proposta deste trabalho.

Um dos maiores desafios ao desenvolver a *skill* foi a forma de aprendizado, em que lugar buscar a informação, pois como o console do desenvolvedor Alexa está em constante desenvolvimento e aprimoramento muitos tutoriais ficam defasados tendo que aprender na tentativa e erro. Isso vale também para a utilização dos outros serviços disponibilizados pela Amazon, além do grande volume de informações úteis se encontrarem apenas no idioma inglês. Pensando nisso se desenvolveu um tutorial do que foi feito em português brasileiro que pode ser encontrando em Rodrigues (2022).

Continuando nesta linha de raciocínio, é possível citar que não existem muitos exemplos de códigos, artigos ou pesquisas científicas referente a este tema e quando existem seriam sobre automação residencial e não sobre jogos, entretenimento ou desenvolvimento de histórias para a interação com Alexa.

A Alexa é assertiva em identificar os comandos do usuário em português brasileiro, mas conforme respostas do questionário alguns tiveram dificuldade ao realizar no idioma inglês estadunidense, já outros não tiveram problemas, isto pode ser devido ao sotaque do usuário ou ao modelo da Echo Show.

As ferramentas utilizadas para a especificação do protótipo atenderam as necessidades e foram fáceis para aprender. O ESP32 custou R\$ 84,43 e atendeu todas as necessidades, que seria a possibilidade de utilizar a plataforma do Arduino IDE para programação e ter conexão Wi-Fi.

Por fim, criou-se um jogo que foi recebido de forma positiva pelos usuários e que conseguiu utilizar outros serviços da Amazon além de integrar com o ESP32, deixando um conteúdo extenso de como desenvolver e o que é possível fazer na criação de *skills*.

4.1 EXTENSÕES

Como possíveis extensões para este trabalho sugere-se:

- a) alterar a forma de conexão do Wi-Fi com o Arduino para se criar uma interface e conseguir se conectar em qualquer rede não sendo necessário informar de forma fixa no código fonte;

- b) receber uma mensagem do Arduino via protocolo MQTT e ativar alguma intenção;
- c) aprimorar a lógica do jogo criando modos, dificuldades ou sistema de fases sendo que cada nível o jogo ficaria mais difícil;
- d) criar a *skill* em outra linguagem de programação disponível e mostrar a diferença na estrutura.

REFERÊNCIAS

- ALEXA DEVELOPERS. **Zero to Hero, Part 1: Alexa Skills Kit Overview**. Youtube, 6 fev. 2022. Disponível em: https://www.youtube.com/playlist?list=PLdZn93YfA_1ZP1WFkz6bm08v3zFfWwfGW. Acesso em 10 ago. 2022.
- 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 CLOUDWATCH. **What is Amazon CloudWatch?** 2022. Disponível em: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>. Acesso em: 12 de ago. 2022.
- AMAZON DYNAMODB. **What is Amazon DynamoDB?** 2022. Disponível em: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>. Acesso em: 12 de ago. 2022.
- AMAZON LAMBDA. **AWS Lambda**, 2021. Disponível em: <https://aws.amazon.com/pt/lambda/>. Acesso em: 29 ago. 2021.
- AMAZON WHITEPAPER. **What is Amazon CloudWatch?** 2022. Disponível em: <https://docs.aws.amazon.com/whitepapers/latest/aws-overview/internet-of-things-services.html#aws-iot-core>. Acesso em: 13 de ago. 2022.
- AMAZON WORKSHOP. **What is ESP32?** 2022. Disponível em: <https://catalog.us-east-1.prod.workshops.aws/workshops/5b127b2f-f879-48b9-9dd0-35aff98c7bbc/en-US/module1/esp32>. Acesso em: 19 de ago. 2022.
- COSTA, Rafael Almeida. **Casa inteligente com recurso a tecnologias open source**. 2018. 23 f. Tese de Mestrado (Sistemas e Tecnologias de Informação para as Organizações) - Centro de Ciências Exatas e Naturais, Instituto Politécnico de Viseu, Viseu.
- 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.
- LUIZ, Bruno Aguiar. **Desafios no desenvolvimento de assistentes pessoais virtuais: um estudo comparativo**. 2019. 41 f. Trabalho de Conclusão de Curso (Curso de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campinas.

MOURA JUNIOR, Anderson David; DÍAZ, Francisco Javier de Obaldia; RIBEIRO, Ivandro da Silva; CRISPIM, Honório Assis Filho. **Automação residencial de baixo custo**, 2020. Trabalho de Conclusão de Curso (Engenharia da Computação) - Centro Universitário de Brasília, Brasília.

NASCIMENTO, Manoel Henrique Reis; WENCELEWSKI, Mauro Henrique da Silva; PAIXÃO, Priscila Reis Soares. Development of an automation system, integrating a virtual assistant and IoT devices. **Journal of Engineering and Technology for Industrial Applications**, v.5, n. 19, p. 69-74, jun./set. 2019.

LOCATELLI, Caroline. **Conhecendo o ESP32**, 2018. Disponível em: <https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>. Acesso em: 19 de ago. 2022.

OLIVEIRA, Euler. **Conhecendo o NodeMCU-32S ESP32**, 2017. Disponível em: <https://blogmasterwalkershop.com.br/embarcados/esp32/conhecendo-o-nodemcu-32s-esp32>. Acesso em: 19 de ago. 2022.

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

RODRIGUES, Rafael dos Santos. **Alexa-Tutorial-PT-BR**, 2022. Disponível em: <https://github.com/rafsarodrigues/Alexa-Tutorial-PT-BR/blob/main/README.md>. Acesso em 05 nov. 2022.

SAGE, Simon. **Guide to smart light switches**, 2021. Disponível em: <https://www.digitaltrends.com/home/quick-guide-to-smart-light-switch/#dt-heading-what-does-a-smart-light-switch-do>. Acesso em: 19 de ago. 2022.

SOHO. **Home Automation and Virtual Assistants – What Are People Using Now?** 2017. Disponível em: <https://soho.com.au/articles/whats-new-home-automation-virtual-assistance>. Acesso em 19 de ago. 2022.

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.

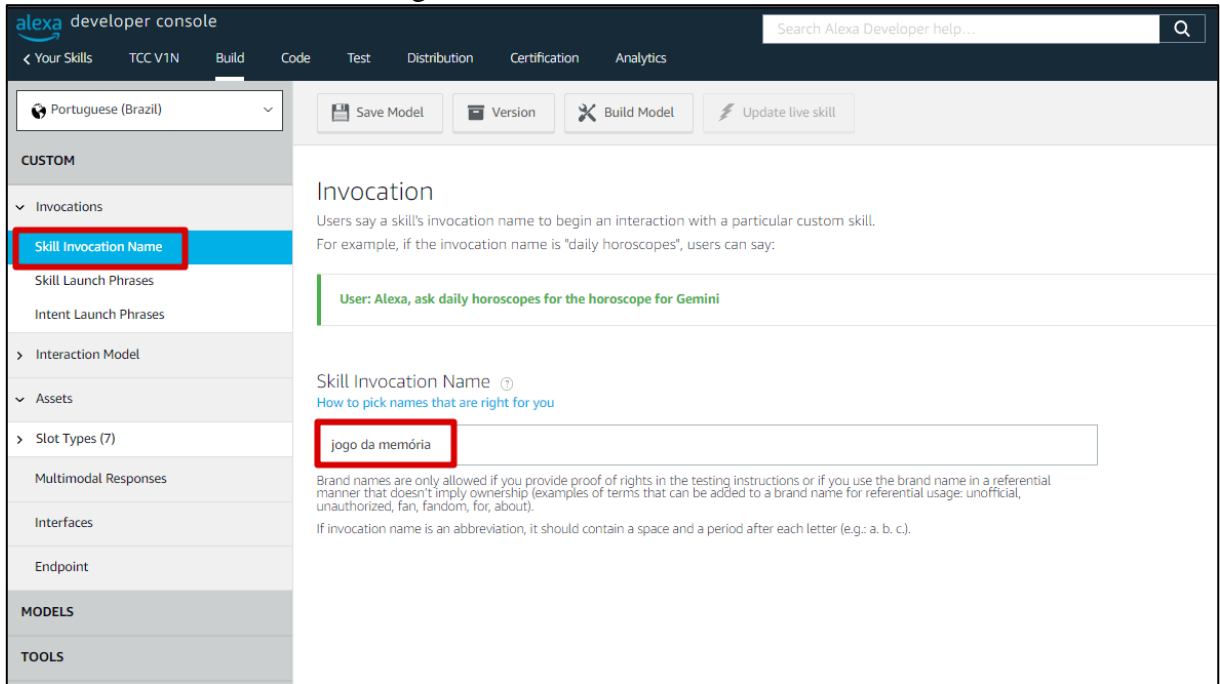
VIGLIAROLO, Brandon. **Amazon Alexa: Cheat Sheet**. [S.I.], 2020. Disponível em: <https://www.techrepublic.com/article/amazon-alexa-the-smart-persons-guide>. Acesso em: 28 ago. 2021.

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.

APÊNDICE A – Como cadastrar itens essenciais para o funcionamento da *skill*

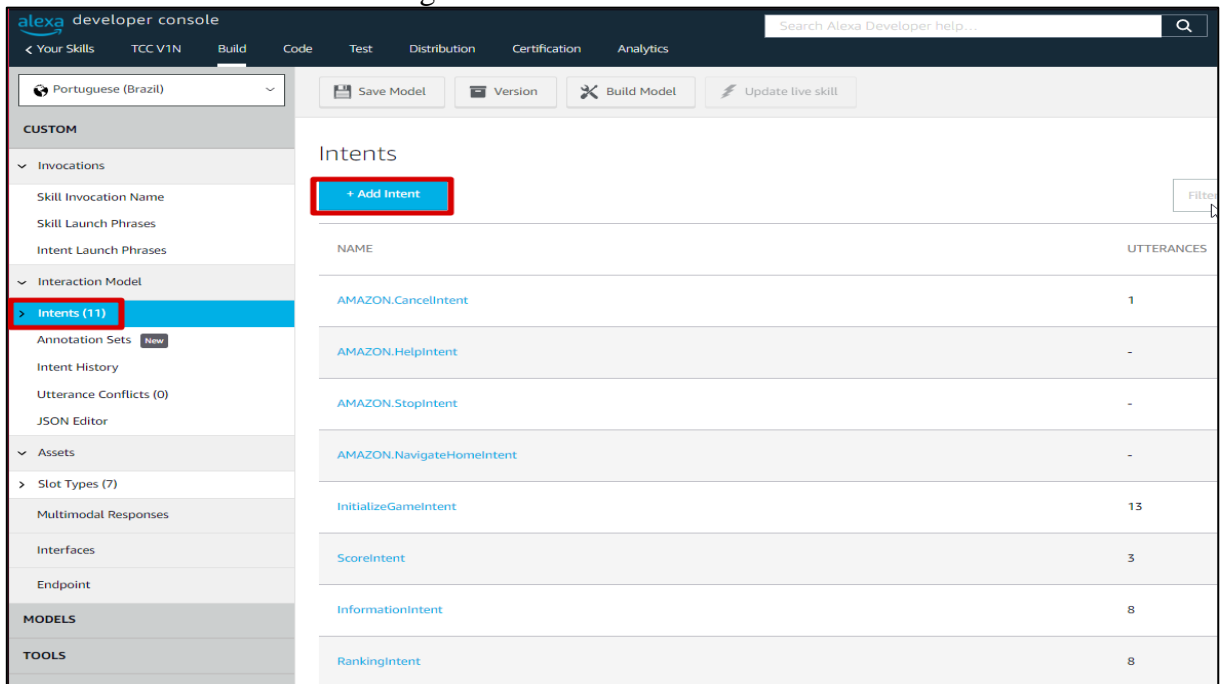
Este apêndice apresenta da Figura 14 - Criar *Invocation Name* Figura 14 a Figura 19 como cadastrar os itens essenciais para o desenvolvimento de uma *skill* Alexa.

Figura 14 - Criar *Invocation Name*

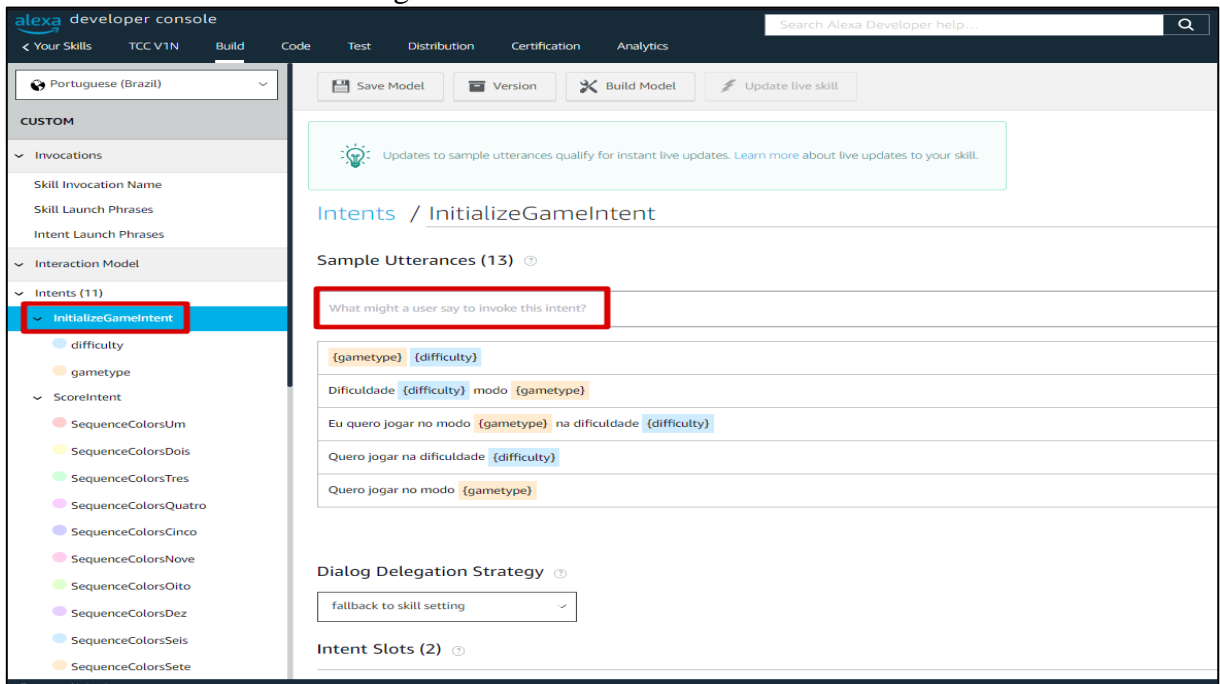


Fonte: elaborado pelo autor.

Figura 15 - Adicionar novo *Intent*

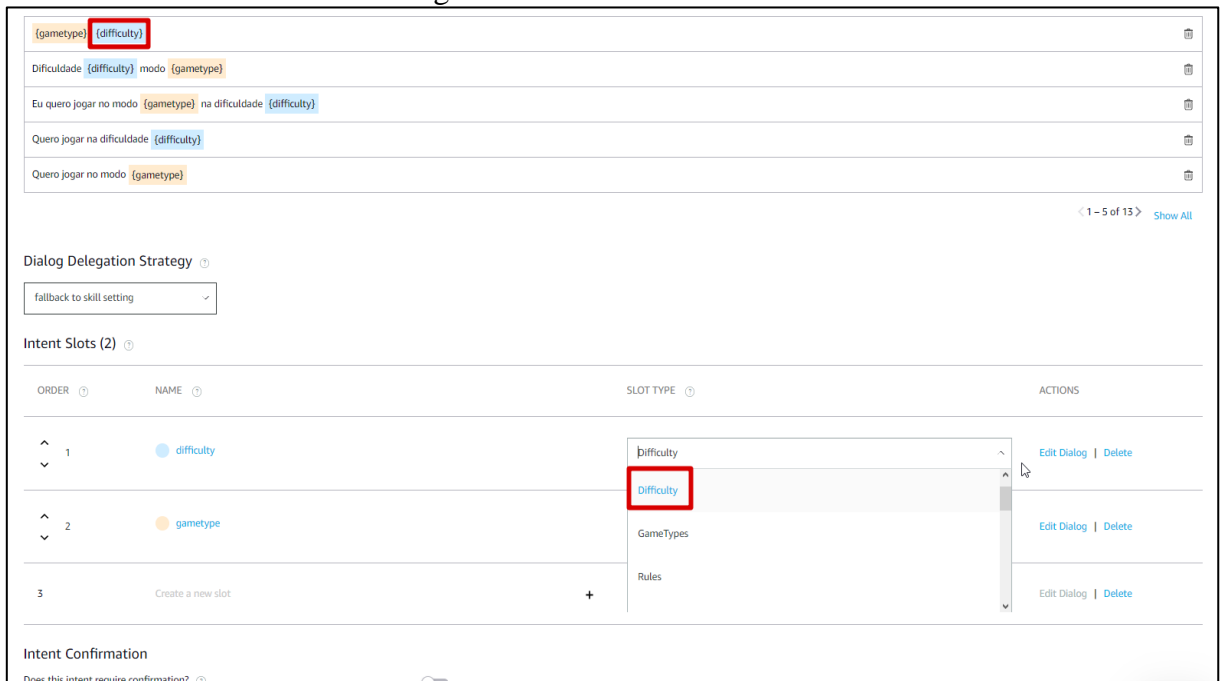


Fonte: elaborado pelo autor.

Figura 16 - Adicionar nova *Utterance*

Fonte: elaborado pelo autor.

Ao utilizar chaves encapsulando uma palavra, automaticamente o sistema irá criar o slot sendo necessário a configuração dele.

Figura 17 - Adicionar novo *slot*

Fonte: elaborado pelo autor.

É possível criar *slots* personalizados, informando palavras que ele deverá reconhecer.

Figura 18 - Configurar *slot type*

The screenshot shows the 'Slot Types' configuration page for the 'Difficulty' slot type. The left sidebar lists various components, with 'Slot Types (7)' highlighted. The main area shows the 'Slot Values (3)' section, which includes a text input field for adding new values and a table of existing values.

VALUE	ID (OPTIONAL)	SYNONYMS (OPTIONAL)	
médio	Enter ID	Add synonym	+
difícil	Enter ID	Add synonym	+
fácil	Enter ID	Add synonym	+

Below the table, the 'Slots Using (Difficulty)' section shows a table with the following data:

SLOT NAME	INTENT
difficulty	InitializeGameIntent

Fonte: elaborado pelo autor.

Figura 19 - Built-In Intents

The screenshot shows the 'Intents' page in the Alexa Developer Console. The left sidebar lists various components, with 'Intents (11)' highlighted. The main area shows a table of built-in intents, with a red box highlighting the first four rows.

NAME	UTTERANCES
AMAZON.CancelIntent	1
AMAZON.HelpIntent	-
AMAZON.StopIntent	-
AMAZON.NavigateHomeIntent	-
InitializeGameIntent	13
ScoreIntent	3
InformationIntent	8
RankingIntent	8
ShowRankingIntent	11

Fonte: elaborado pelo autor.

APÊNDICE B – Descrição dos Casos de Uso

Este apêndice descreve os casos de uso abordados nos diagramas expostos na seção 3.2.2 deste trabalho. No Quadro 14 apresenta-se a descrição do caso de uso iniciar a *skill*.

Quadro 14 - Descrição do UC01

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.

Fonte: elaborado pelo autor.

No Quadro 15 apresenta-se a descrição do caso de uso iniciar o jogo.

Quadro 15 - Descrição UC02

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.

Fonte: elaborado pelo autor.

No Quadro 16 apresenta-se a descrição do caso de uso enviar JSON via protocolo MQTT.

Quadro 16 - Descrição do UC03

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.

Fonte: elaborado pelo autor.

No Quadro 17 apresenta-se a descrição do caso de uso responder questão.

Quadro 17 - Descrição do UC04

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.

Fonte: elaborado pelo autor.

No Quadro 18 apresenta-se a descrição do caso de uso registrar pontuação.

Quadro 18 - Descrição do UC05

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.

Fonte: elaborado pelo autor.

No Quadro 19 apresenta-se a descrição do caso de uso atualizar pontuação.

Quadro 19 - Descrição do UC06

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.

Fonte: elaborado pelo autor.

No Quadro 20 apresenta-se a descrição do caso de uso consultar regras.

Quadro 20 - Descrição do UC07

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.

Fonte: elaborado pelo autor.

No Quadro 21 apresenta-se a descrição do caso de uso consultar informações.

Quadro 21 - Descrição do UC08

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.

Fonte: elaborado pelo autor.

No Quadro 22 apresenta-se a descrição do caso de uso consultar pontuações.

Quadro 22 - Descrição do UC09

UC09 – Consultar pontuações

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:

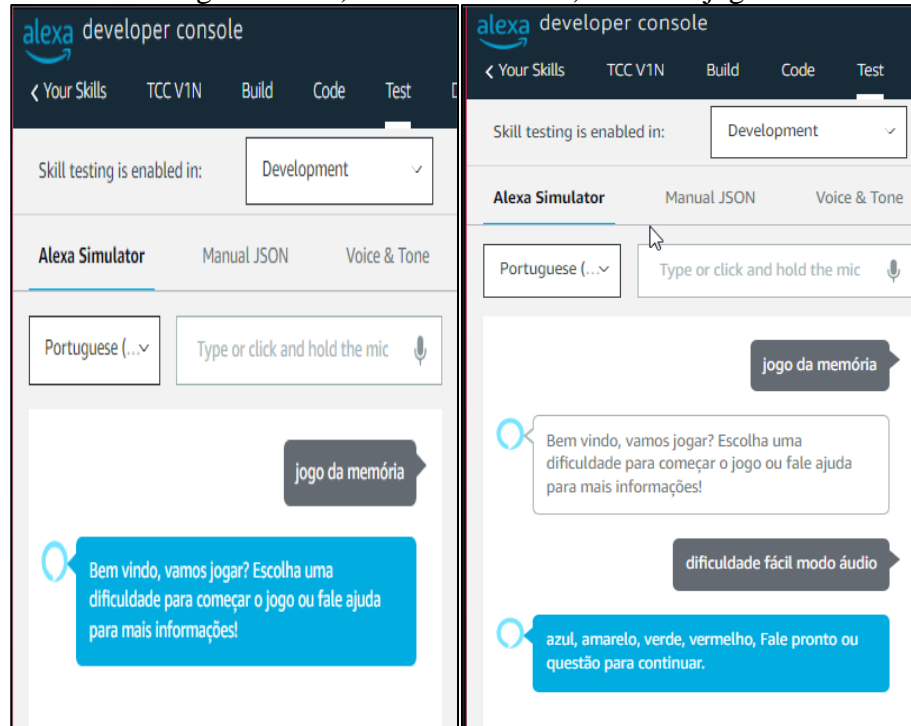
1. Usuário utiliza o comando para requisitar a lista de pontuação;
2. A Alexa reconhece o comando;
3. Alexa fala a lista de pontuação dos três primeiros colocados.

Fonte: elaborado pelo autor.

APÊNDICE C – Execução dos comandos no console do desenvolvedor Alexa

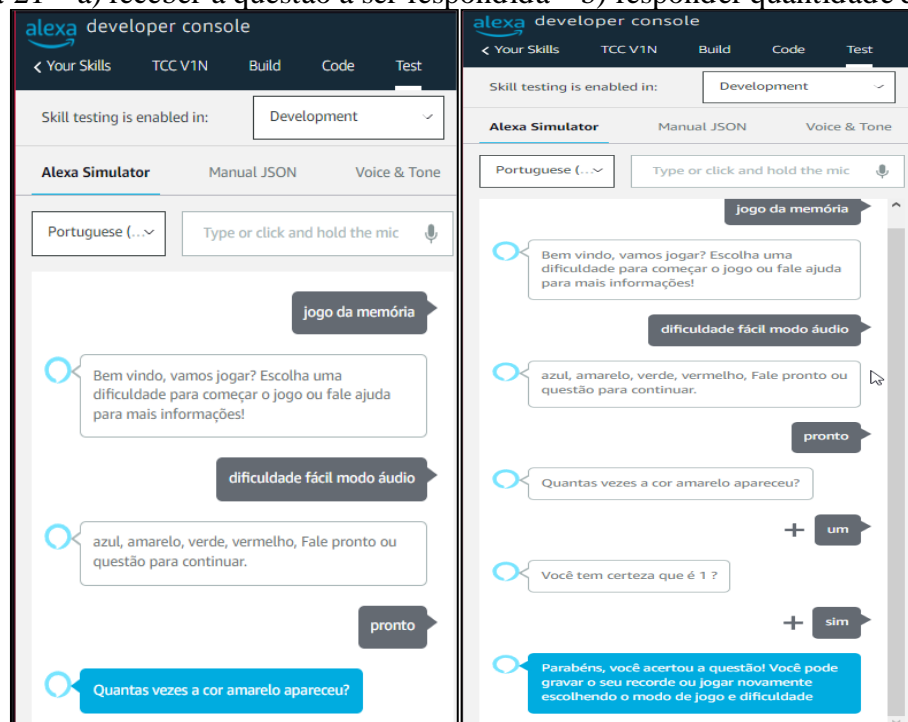
Este apêndice apresenta os comandos citados nos quadros Quadro 9 e Quadro 10, da Figura 20 à Figura 25 os comandos em português brasileiro. Já da Figura 26 à Figura 30 em inglês estadunidense.

Figura 20 – a) iniciar a *skill* – b) iniciar o jogo



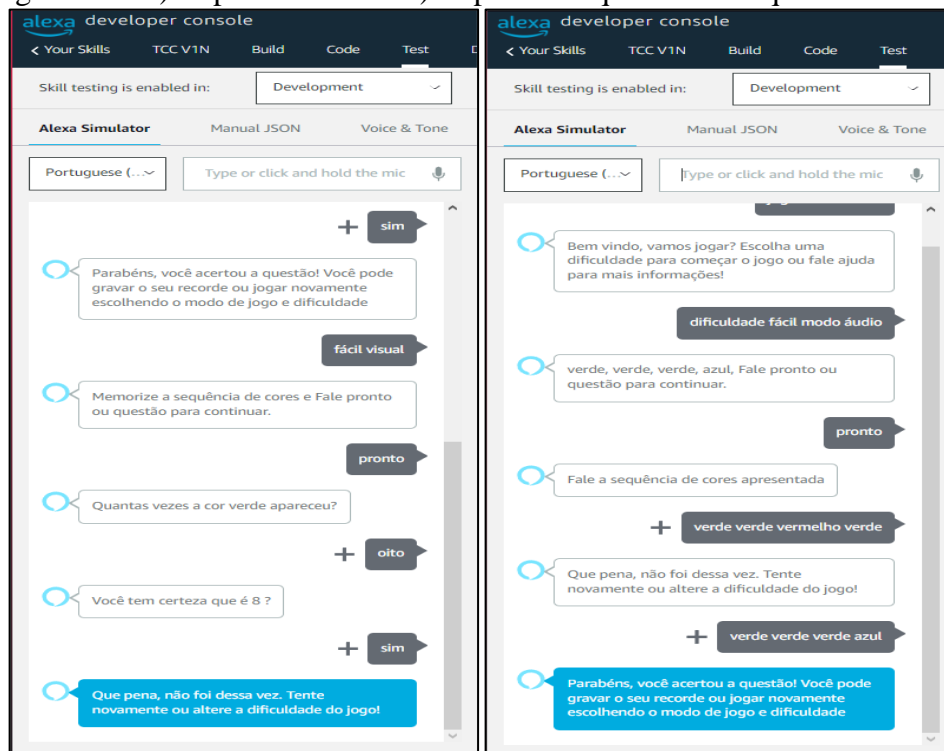
Fonte: elaborado pelo autor.

Figura 21 – a) receber a questão a ser respondida – b) responder quantidade de cores



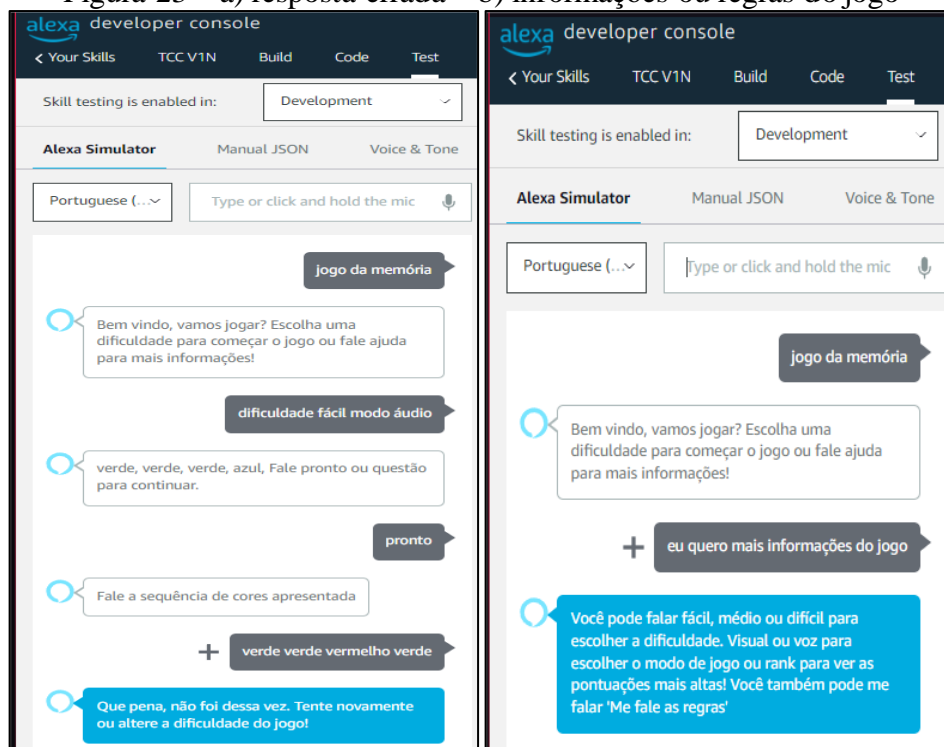
Fonte: elaborado pelo autor.

Figura 22 – a) resposta errada – b) responder à questão de sequência de cores



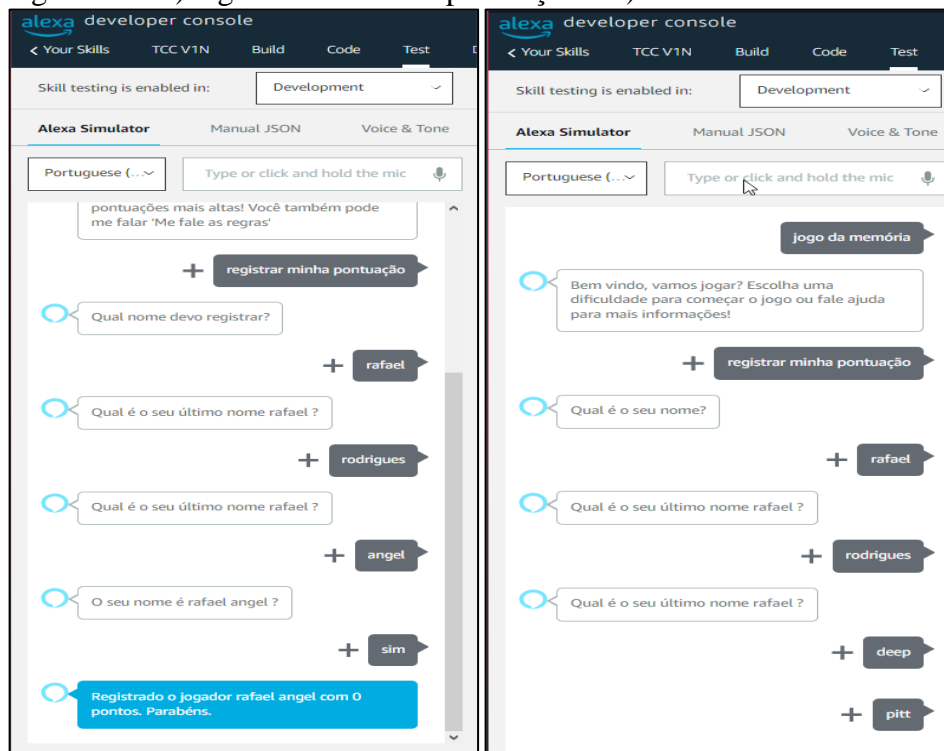
Fonte: elaborado pelo autor.

Figura 23 – a) resposta errada – b) informações ou regras do jogo



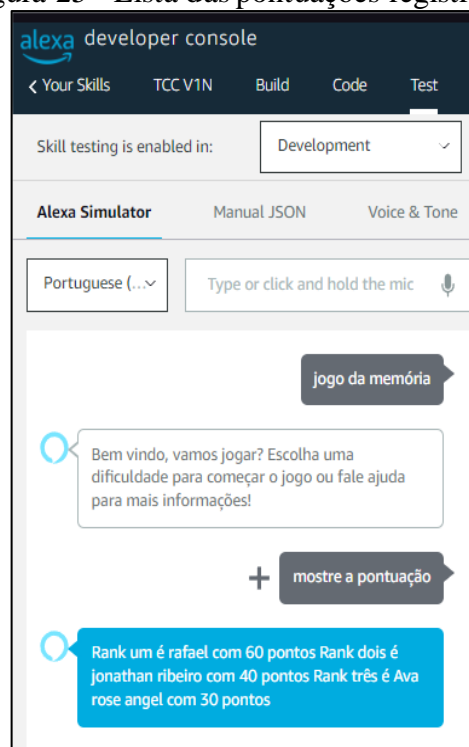
Fonte: elaborado pelo autor.

Figura 24 – a) registrar ou excluir pontuação – b) nomes não considerados



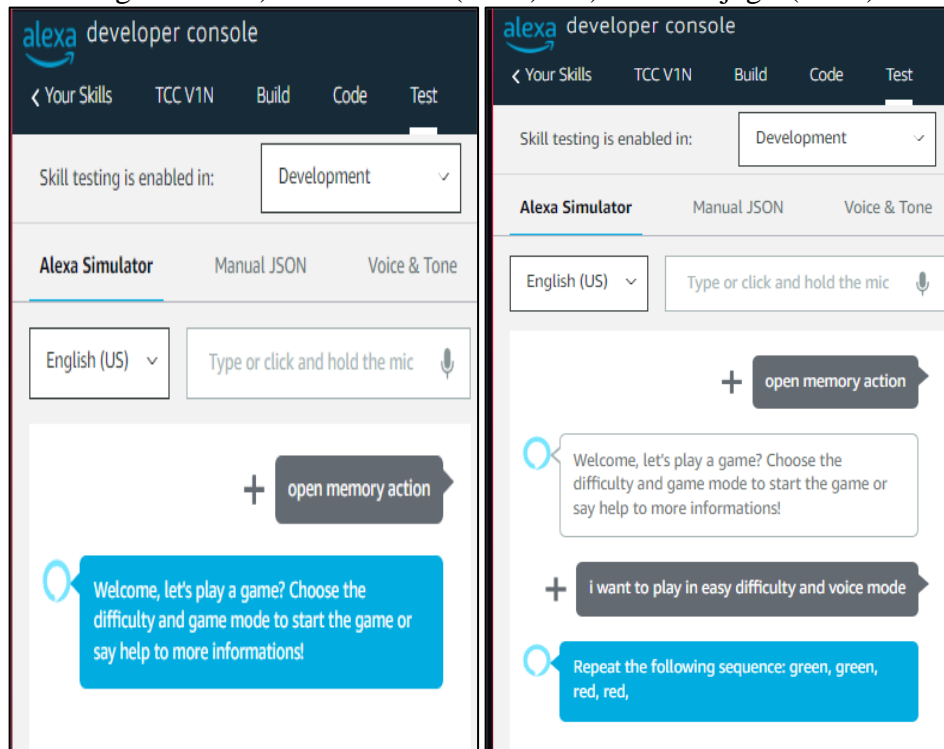
Fonte: elaborado pelo autor.

Figura 25 - Lista das pontuações registradas



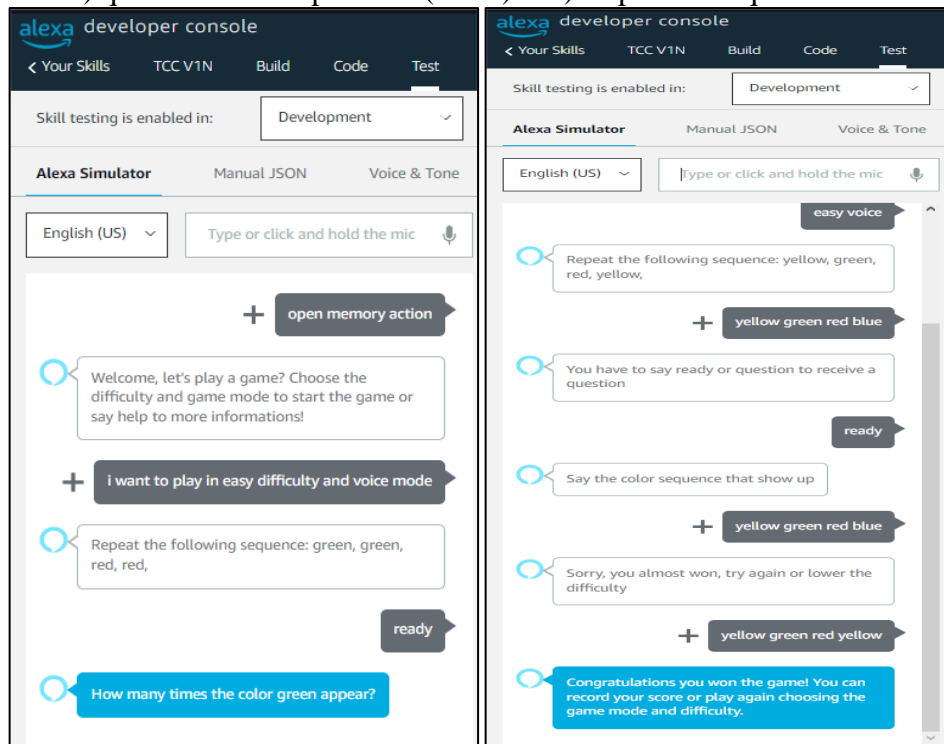
Fonte: elaborado pelo autor.

Figura 26 – a) iniciar a *skill* (en-us) – b) iniciar o jogo (en-us)



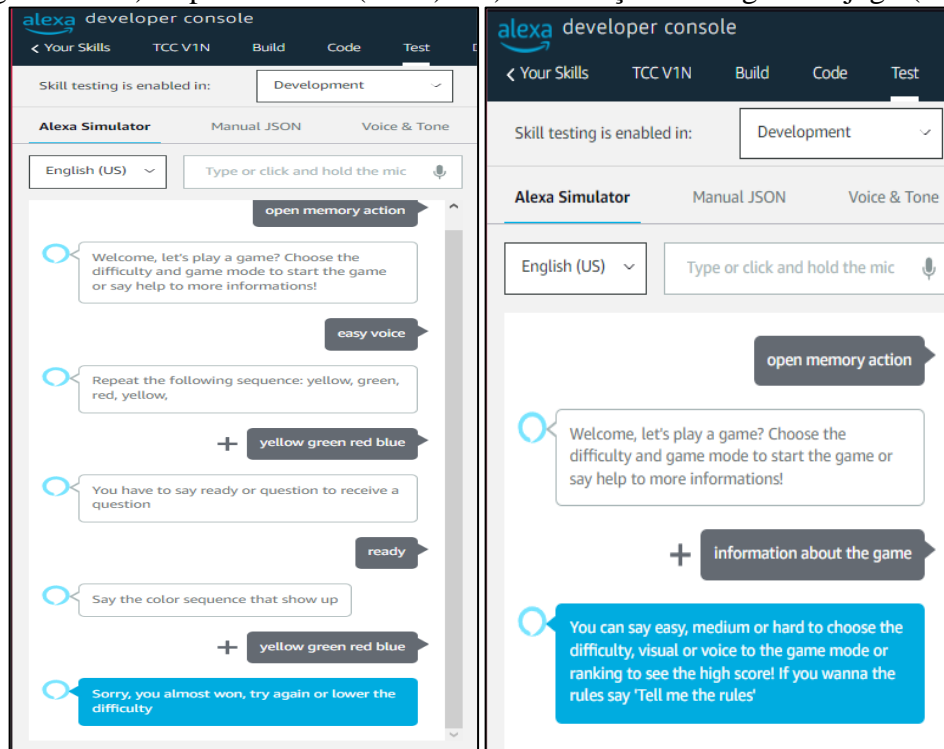
Fonte: elaborado pelo autor.

Figura 27 – a) questão a ser respondida (en-us) – b) responder sequência de cores (en-us)



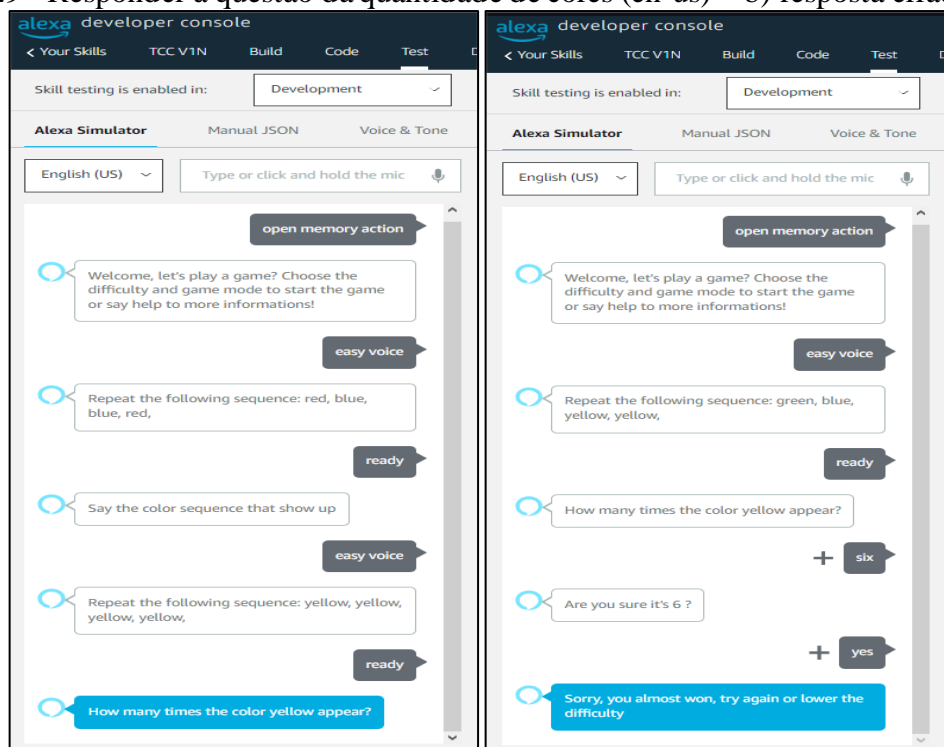
Fonte: elaborado pelo autor.

Figura 28 – a) resposta errada (en-us) – b) informações ou regras do jogo (en-us)



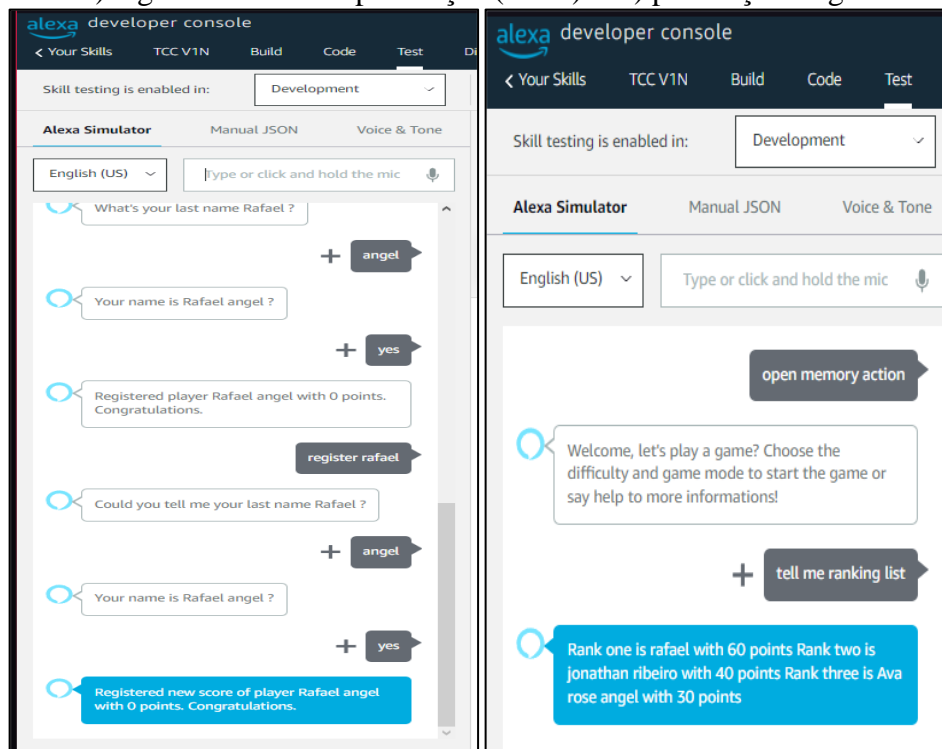
Fonte: elaborado pelo autor.

Figura 29 - Responder à questão da quantidade de cores (en-us) – b) resposta errada (en-us)



Fonte: elaborado pelo autor.

Figura 30 – a) registrar ou exclui pontuação (en-us) – b) pontuações registradas (en-us)




Fonte: elaborado pelo autor.

APÊNDICE D – Roteiro de avaliação da *skill*

Este apêndice apresenta o roteiro de avaliação e o questionário utilizado para a avaliação da *skill* das figuras Figura 31 à Figura 43.

Figura 31 - Primeira seção questionário



Skill - Jogo da memória

Este questionário tem como objetivo avaliar a usabilidade e o desempenho do Trabalho de Conclusão de Curso "JOGO DA MEMÓRIA: DESENVOLVIMENTO DE SKIL COM CONCEITOS DO JOGO GENIUS", do aluno Rafael dos Santos Rodrigues no Curso de Sistemas de Informação, FURB, com o Orientador Professor Dalton Solano dos Reis.

[Faça login no Google](#) para salvar o que você já preencheu. [Saiba mais](#)

[Próxima](#) Página 1 de 5 [Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários

Fonte: elaborado pelo autor.

Figura 32 - Segunda seção questionário

Perfil de Usuário

Nome *

Sua resposta

Qual o seu grau de escolaridade? *

☐ Ensino fundamental incompleto

☐ Ensino fundamental completo – 1º grau

☐ Ensino médio incompleto

☐ Ensino médio completo – 2º grau

☐ Ensino superior incompleto

☐ Ensino superior completo

Você conhece o que é uma assistente virtual? *

☐ Sim, conheço e já utilizei

☐ Sim, mas só ouvi falar, não tenho muito conhecimento sobre

☐ Não, nunca ouvi falar

Fonte: elaborado pelo autor.

Figura 33 - Segunda seção questionário (parte 2)

Você utiliza alguma assistente virtual? Se sim, qual?

☐ Alexa

☐ Google

☐ Siri

☐ Cortana

☐ Outro: _____

Com qual frequência você utiliza uma assistente virtual?

☐ Frequentemente

☐ Às vezes

☐ Nunca utilizei

Fonte: elaborado pelo autor.

Figura 34 - Segunda seção questionário (parte 3)

Para quê você utiliza a assistente virtual?

☐ Música

☐ Automação residencial

☐ Jogos

☐ Compras

☐ Receitas

☐ Vídeos

☐ Outro: _____

Fonte: elaborado pelo autor.

Figura 35 - Terceira seção questionário

Aprendendo a utilizar a skill "Jogo da Memória"

Neste primeiro momento a ideia é você explorar os comandos que podem ser feitos dentro da skill, quais as regras, modo de jogo, como se registrar/excluir.

Abra a skill com o comando "Alexa, abrir jogo da memória". *

☐ Concluído

☐ Não conseguiu concluir

Utilize o comando "Alexa, eu quero mais informações do jogo". *

☐ Concluído

☐ Não conseguiu concluir

Utilize o comando "Alexa, regras do jogo". *

☐ Concluído

☐ Não conseguiu concluir

Fonte: elaborado pelo autor.

Figura 36 - Terceira seção questionário (parte 2)

A skill oferece dois modos de jogos, áudio ou visual. Caso escolha áudio, a Alexa *
irá falar uma lista de cores. Caso escolha visual, a Alexa irá enviar um comando
ao aparelho ligar/desligar as lâmpadas de forma sequencial.
O jogo conta com 2 dificuldades: fácil e médio.
Utilize o comando "Alexa, dificuldade fácil/médio modo visual/voz".

- ☐ Concluído
- ☐ Não conseguiu concluir

Após realizar o comando escolhendo o modo e dificuldade do jogo, deve falar *
"Alexa, pronto" para receber uma questão.

- ☐ Concluído
- ☐ Não conseguiu concluir

Após receber a questão responda com o que é pedido, sendo um valor numérico *
ou a sequência de cores.

- ☐ Concluído
- ☐ Não conseguiu concluir

Fonte: elaborado pelo autor.

Figura 37 - Terceira seção questionário (parte 3)

Após ganhar uma partida é possível cadastrar a sua pontuação. Para isso utilize *
o comando "Alexa, registrar minha pontuação".

Quando Alexa questionar o seu sobrenome é necessário utilizar um nome americano ou novamente o seu primeiro nome.

☐ Concluído

☐ Não conseguiu concluir

Para saber as maiores pontuações, utilize o comando "Mostre a pontuação". *

☐ Concluído

☐ Não conseguiu concluir

Fonte: elaborado pelo autor.

Figura 38 - Quarta seção questionário

Jogar em inglês estadunidense

Caso desejar, a skill conta com a opção de jogar em inglês além das dificuldades fácil e médio, também consta com a dificuldade difícil.

Alexa, open memory action. *

☐ Concluído

☐ Não conseguiu concluir

Alexa, information about the game. *

☐ Concluído

☐ Não conseguiu concluir

Fonte: elaborado pelo autor.

Figura 39 - Quarta seção questionário (parte 2)

<p>Alexa, tell me the rules. *</p> <p><input type="radio"/> Concluído</p> <p><input type="radio"/> Não conseguiu concluir</p>
<p>Alexa, I want to play in easy/medium/hard difficulty and visual/voice mode. *</p> <p><input type="radio"/> Concluído</p> <p><input type="radio"/> Não conseguiu concluir</p>
<p>Alexa, ready. *</p> <p><input type="radio"/> Concluído</p> <p><input type="radio"/> Não conseguiu concluir</p>
<p>Após receber a questão responda com o que é pedido, sendo um valor numérico * ou a sequência de cores.</p> <p><input type="radio"/> Concluído</p> <p><input type="radio"/> Não conseguiu concluir</p>

Fonte: elaborado pelo autor.

Figura 40 - Quarta seção questionário (parte 3)

Alexa, register my points. *

Quando Alexa questionar o seu sobrenome é necessário utilizar um nome americano ou novamente o seu primeiro nome.

☐ Concluído

☐ Não conseguiu concluir

Alexa, tell the ranking list. *

☐ Concluído

☐ Não conseguiu concluir

Fonte: elaborado pelo autor.

Figura 41 - Quinta seção questionário

Avaliação Geral

Você conseguiu concluir os objetivos dessa pesquisa com facilidade? *

1 2 3 4 5

Nenhuma foi fácil ☐ ☐ ☐ ☐ ☐ Todas foram fáceis

Quantas tarefas você concluiu sem NENHUM auxílio externo? *

☐ Nenhuma

☐ Menos que três

☐ Todas

Fonte: elaborado pelo autor.

Figura 42 - Quinta seção questionário (parte 2)

Você acredita que pessoas com necessidades especiais(cegos e baixa visão) se beneficiariam com esta proposta de interface diferente, usando skill? *

1 2 3 4 5

Péssima ☐ ☐ ☐ ☐ ☐ Excelente

Como você classifica a usabilidade da Skill - Jogo da Memória? *

1 2 3 4 5

Péssima ☐ ☐ ☐ ☐ ☐ Excelente

Você acha que a skill cumpriu o objetivo de ser um jogo simples com diferencial da assistente virtual? *

0 1 2 3 4 5

Não concordo ☐ ☐ ☐ ☐ ☐ ☐ Concordo Plenamente

Fonte: elaborado pelo autor.

Figura 43 - Quinta seção questionário (parte 3)

Você acha que faltou mais modos de jogo? *

0 1 2 3 4 5

Não concordo ☐ ☐ ☐ ☐ ☐ ☐ Concordo Plenamente

Você possui algum comentário geral, crítica ou sugestão?

Sua resposta _____

Você encontrou algum erro ou melhoria que pode ser feito no aplicativo?

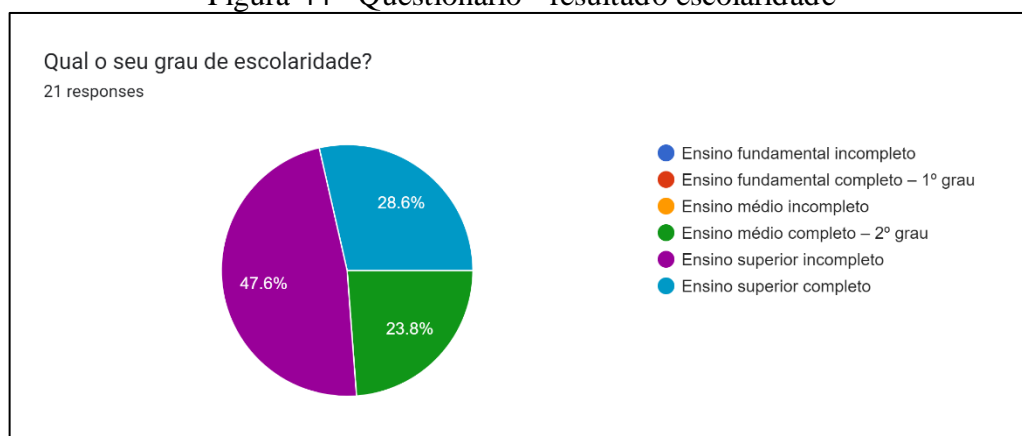
Sua resposta _____

Fonte: elaborado pelo autor.

APÊNDICE E – Perguntas e respostas obtidas na avaliação utilizando questionário

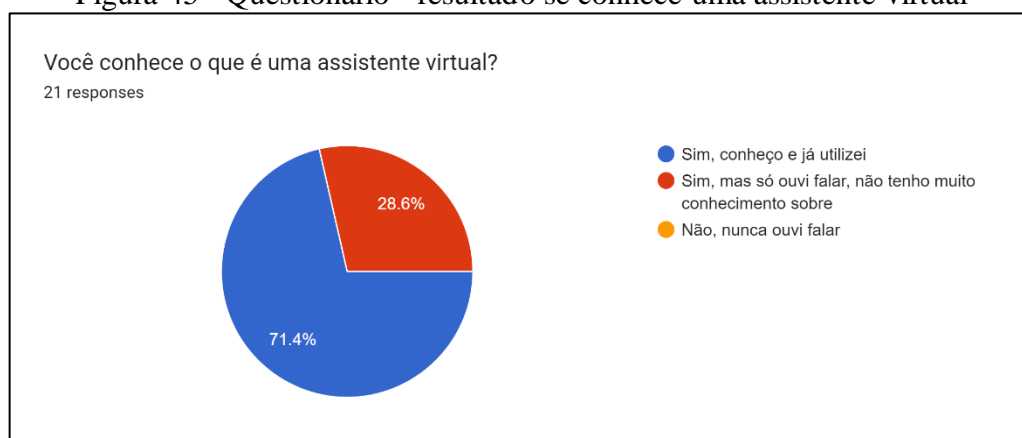
Este apêndice apresenta as perguntas que foram realizadas aos usuários por meio de uma avaliação on-line. Da Figura 44 à Figura 48 apresentam as perguntas e respostas referentes a definição do perfil do usuário. Entre as Figura 48 e Figura 56 se encontra se o usuário conseguiu executar o comando em português brasileiro e já da Figura 57 a Figura 64 se conseguiram realizar os comandos em inglês estadunidense. Por fim, da Figura 65 a Figura 73 as repostas de classificação da *skill*, sugestões, críticas ou possíveis melhorias na *skill*.

Figura 44 - Questionário - resultado escolaridade



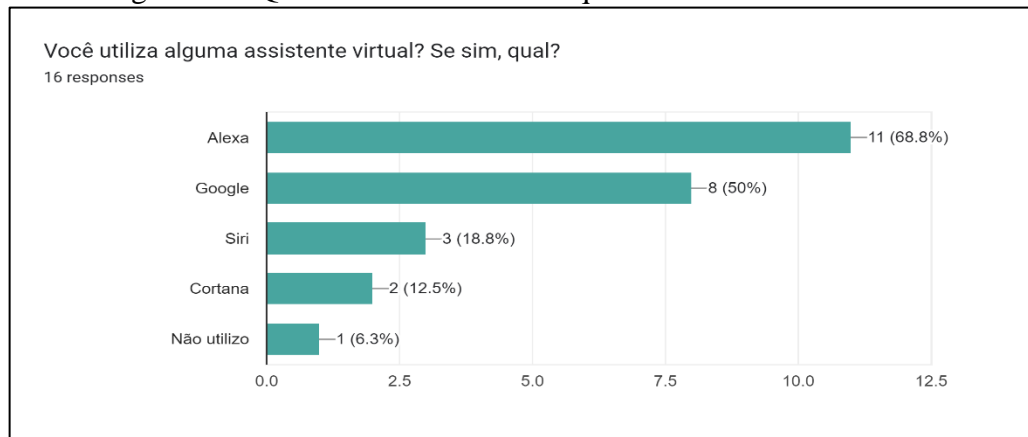
Fonte: elaborado pelo autor.

Figura 45 - Questionário - resultado se conhece uma assistente virtual



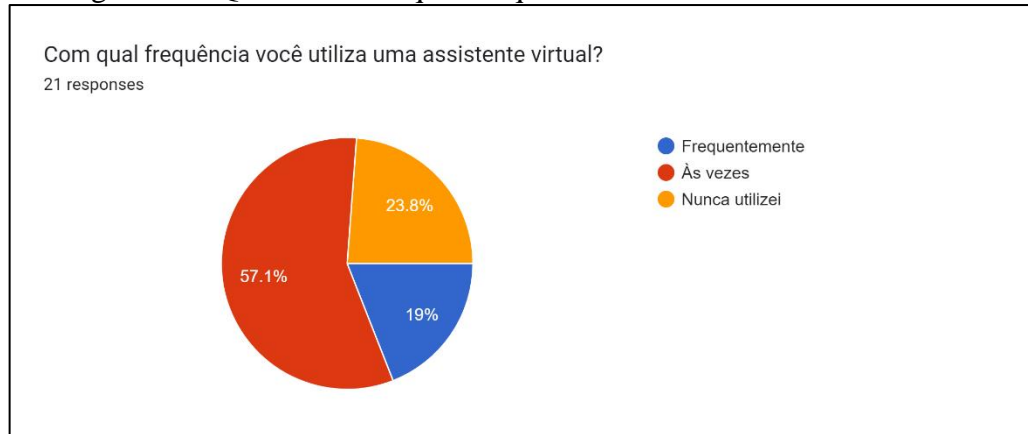
Fonte: elaborado pelo autor.

Figura 46 - Questionário - resultado qual assistente virtual utiliza



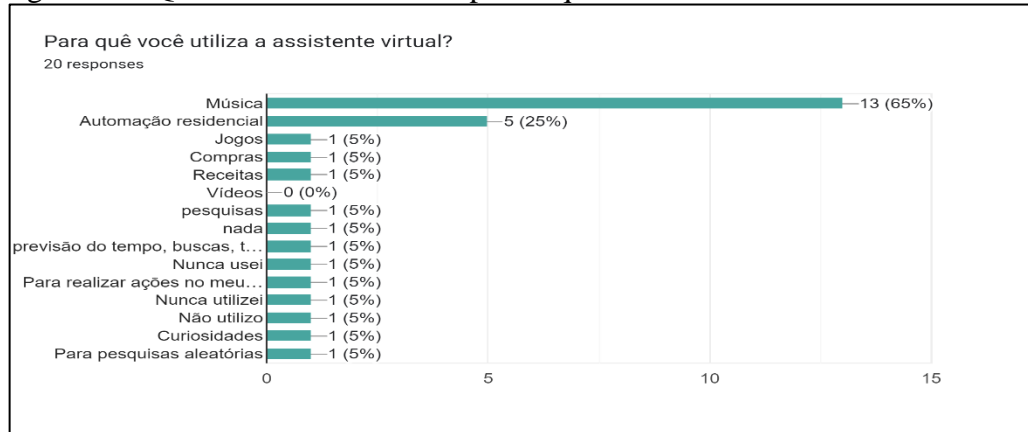
Fonte: elaborado pelo autor.

Figura 47 - Questionário - qual frequência utiliza uma assistente virtual

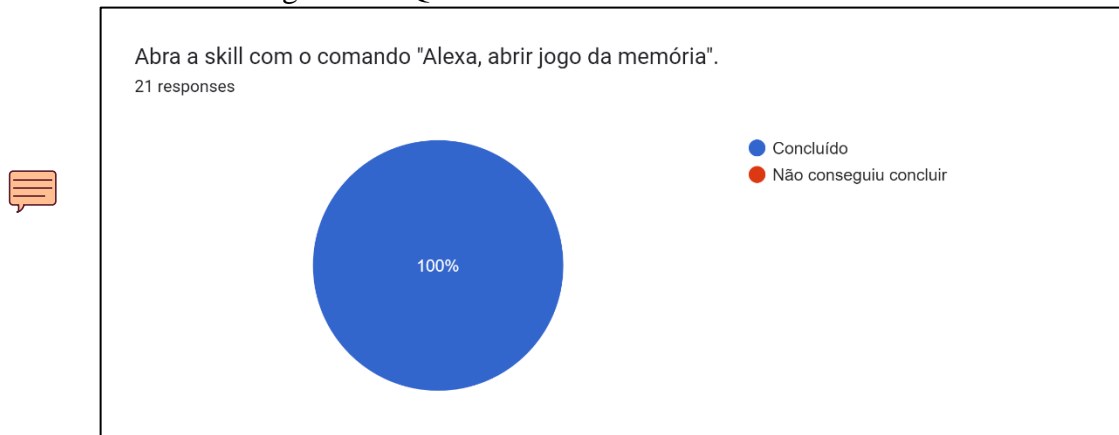


Fonte: elaborado pelo autor.

Figura 48 - Questionário - resultado para o quê você utiliza uma assistente virtual

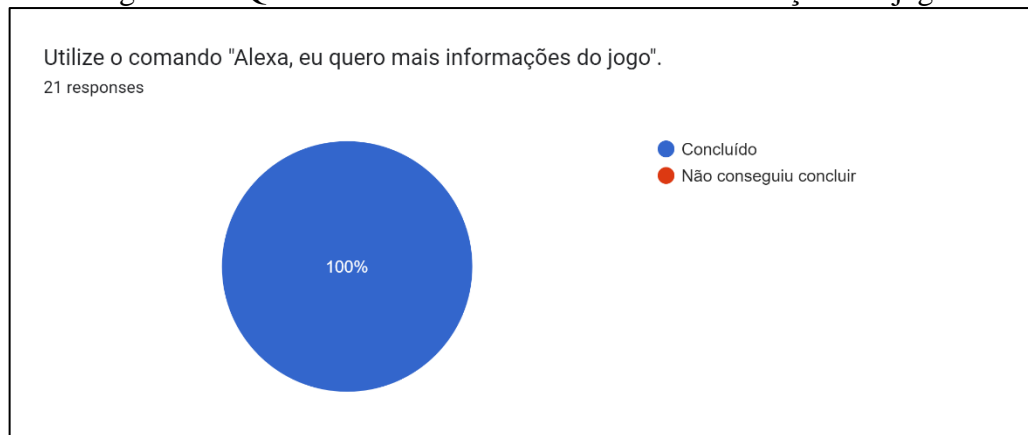


Fonte: elaborado pelo autor.

Figura 49 - Questionário - resultado comando abrir *skill*

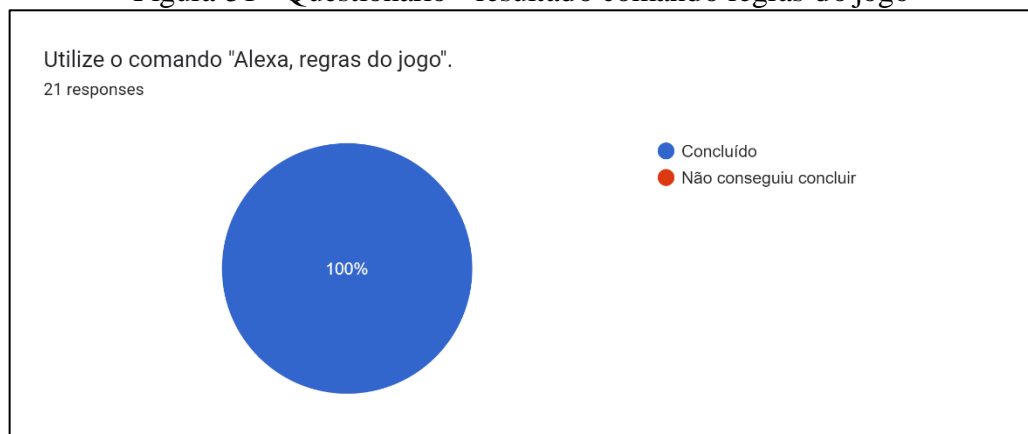
Fonte: elaborado pelo autor.

Figura 50 - Questionário - resultado comando informações do jogo



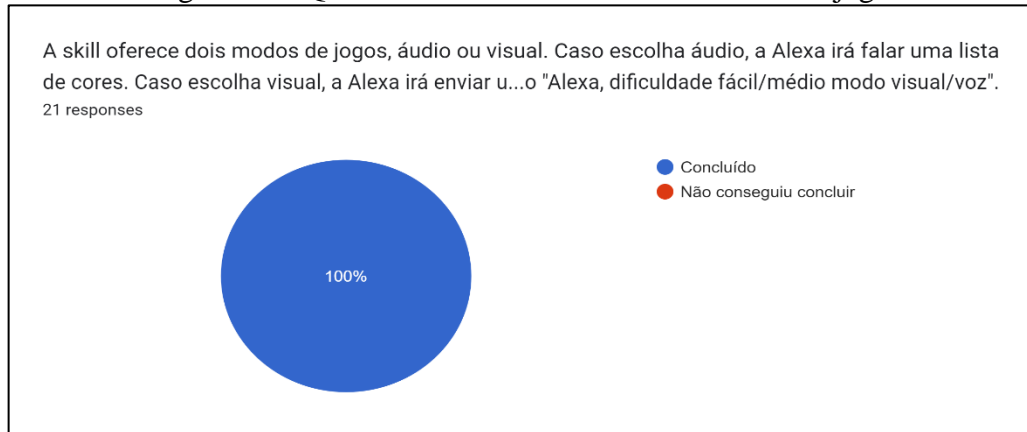
Fonte: elaborado pelo autor.

Figura 51 - Questionário - resultado comando regras do jogo



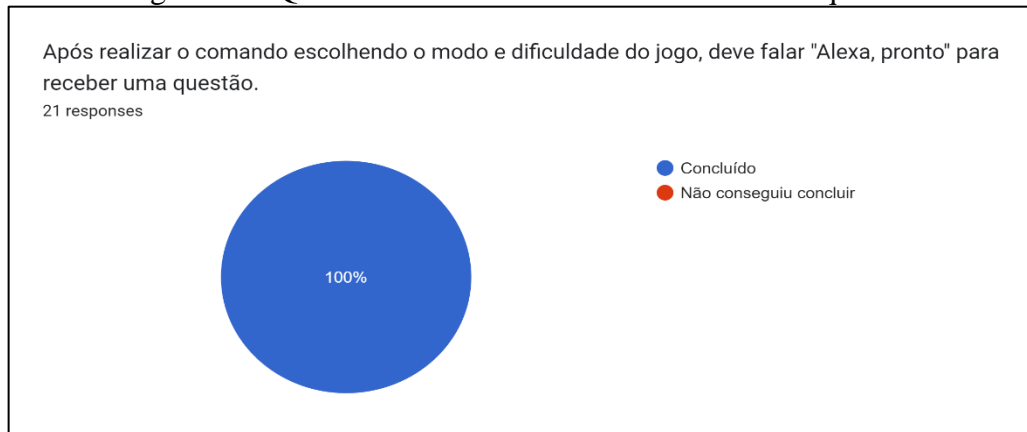
Fonte: elaborado pelo autor.

Figura 52 - Questionário - resultado comando iniciar jogo



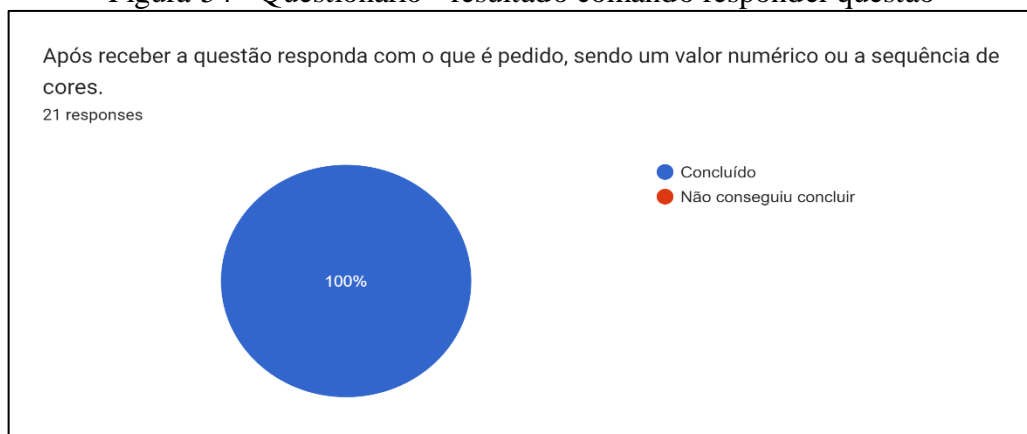
Fonte: elaborado pelo autor.

Figura 53 - Questionário - resultado comando receber questão



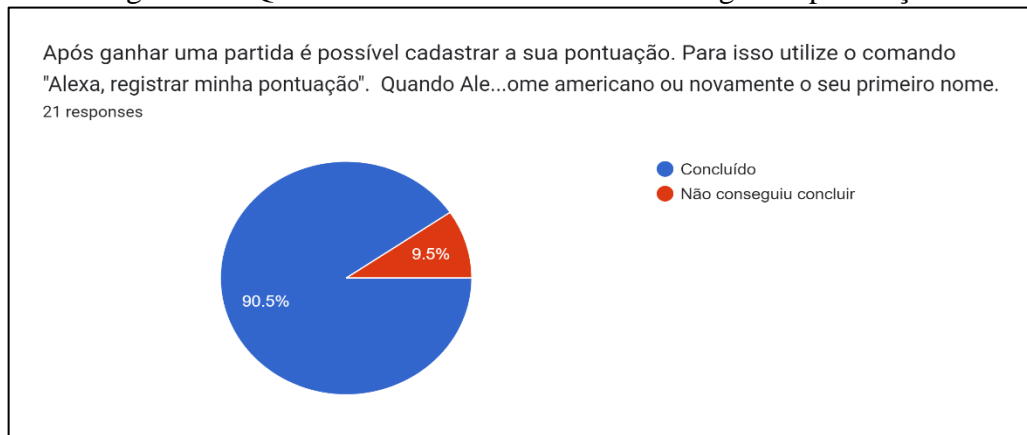
Fonte: elaborado pelo autor.

Figura 54 - Questionário - resultado comando responder questão



Fonte: elaborado pelo autor.

Figura 55 - Questionário - resultado comando registrar pontuação



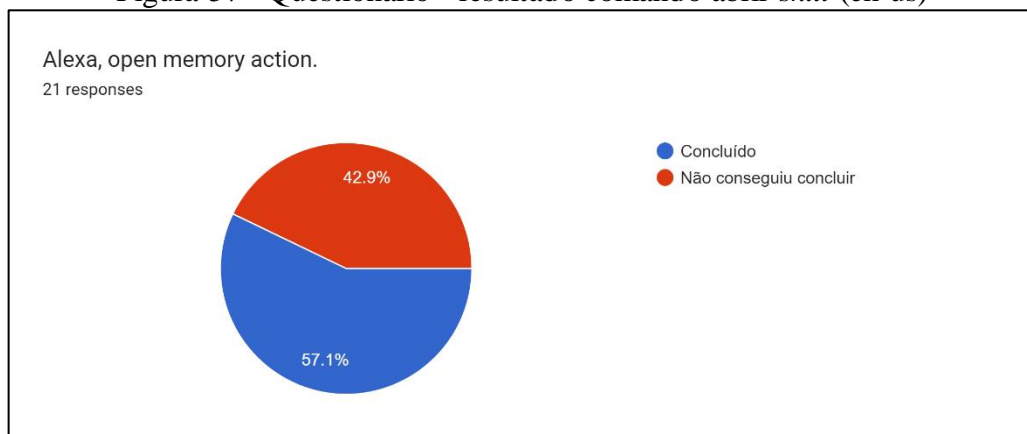
Fonte: elaborado pelo autor.

Figura 56 - Questionário - resultado comando falar lista de pontuação



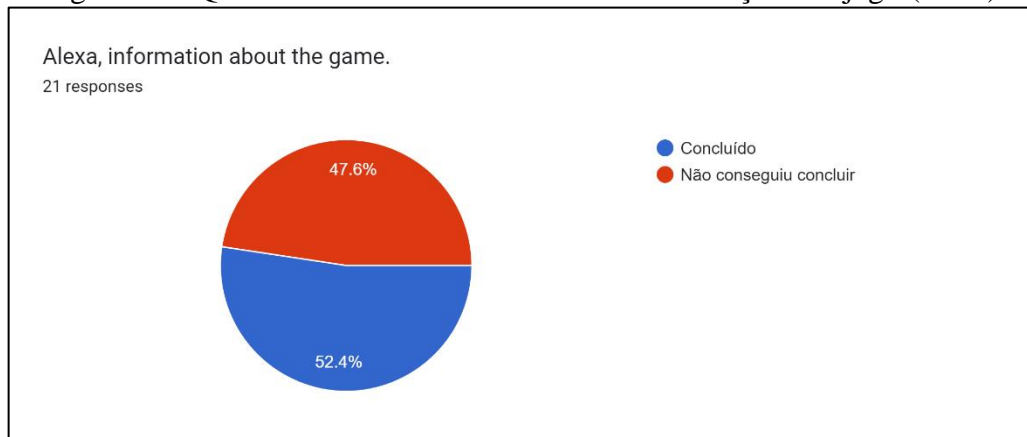
Fonte: elaborado pelo autor.

Figura 57 - Questionário - resultado comando abrir *skill* (en-us)



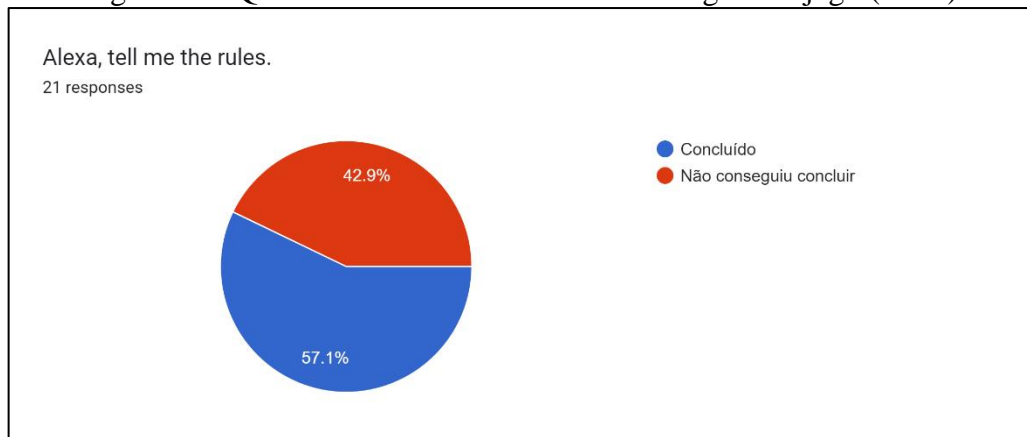
Fonte: elaborado pelo autor.

Figura 58 - Questionário - resultado comando informações do jogo (en-us)



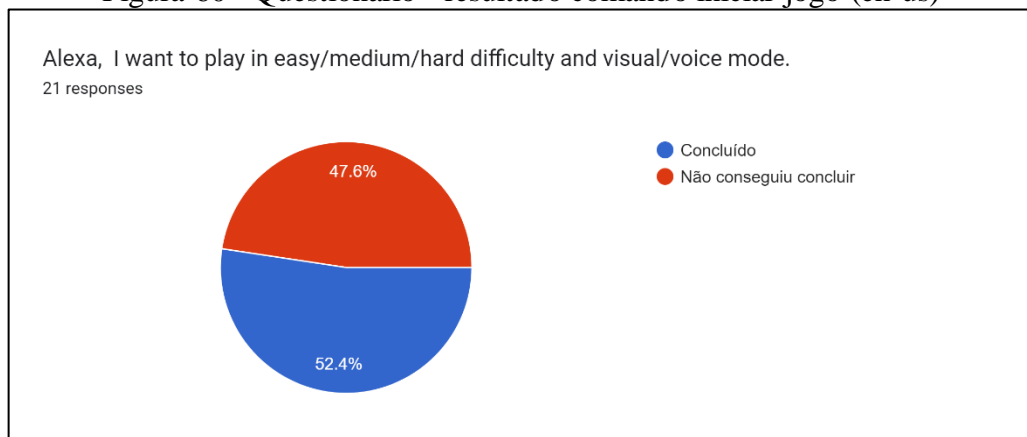
Fonte: elaborado pelo autor.

Figura 59 - Questionário - resultado comando regras do jogo (en-us)



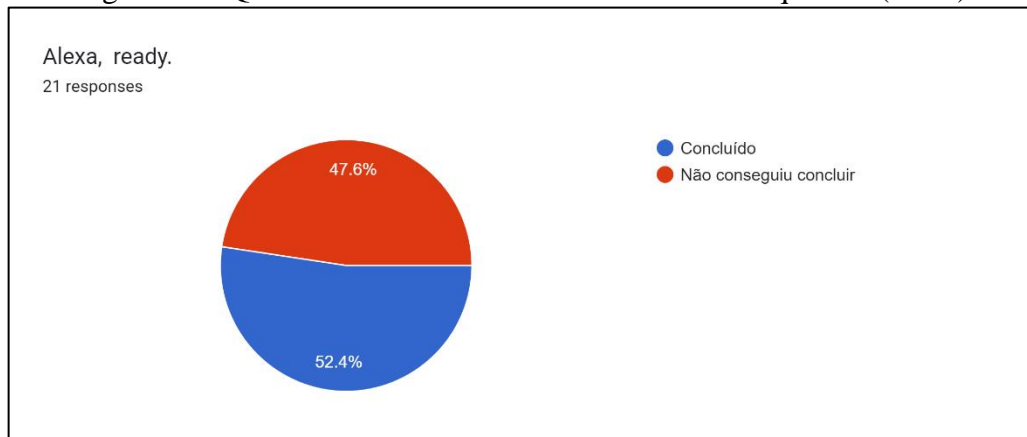
Fonte: elaborado pelo autor.

Figura 60 - Questionário - resultado comando iniciar jogo (en-us)



Fonte: elaborado pelo autor.

Figura 61 - Questionário - resultado comando receber questão (en-us)



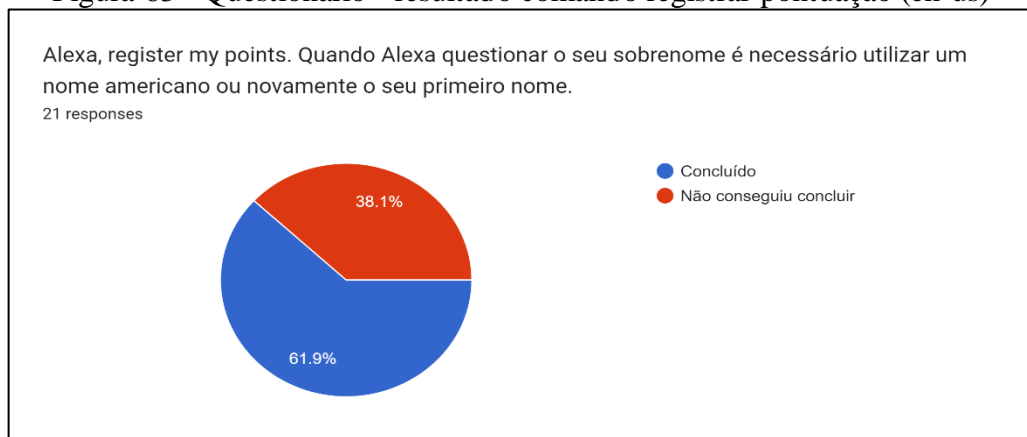
Fonte: elaborado pelo autor.

Figura 62 - Questionário - resultado comando responder questão (en-us)



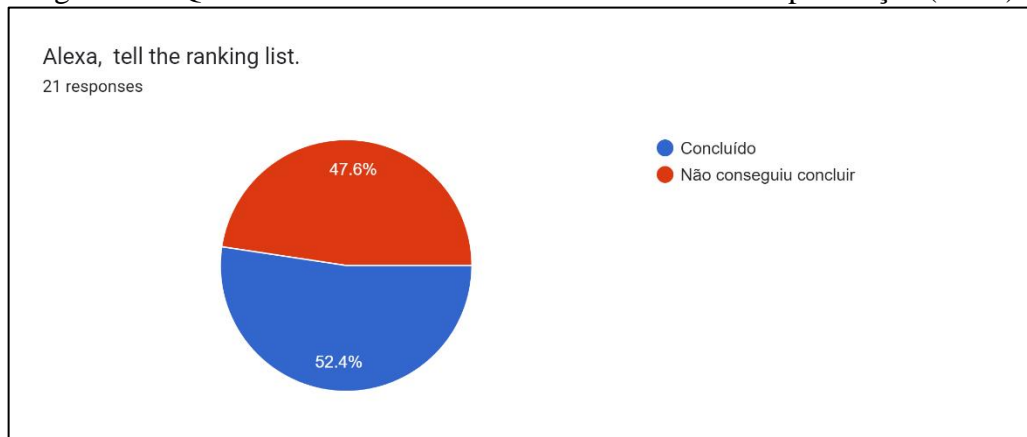
Fonte: elaborado pelo autor.

Figura 63 - Questionário - resultado comando registrar pontuação (en-us)



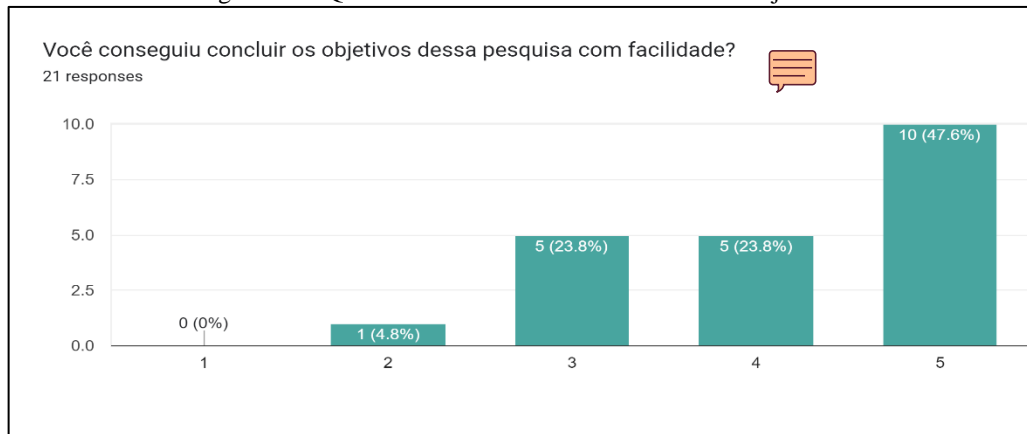
Fonte: elaborado pelo autor.

Figura 64 - Questionário - resultado comando falar lista de pontuação (en-us)



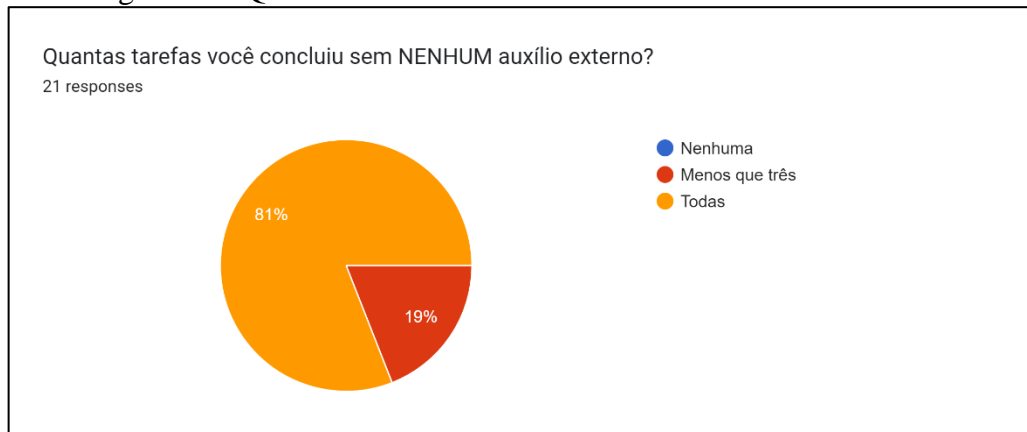
Fonte: elaborado pelo autor.

Figura 65 - Questionário - resultado conclusão dos objetivos



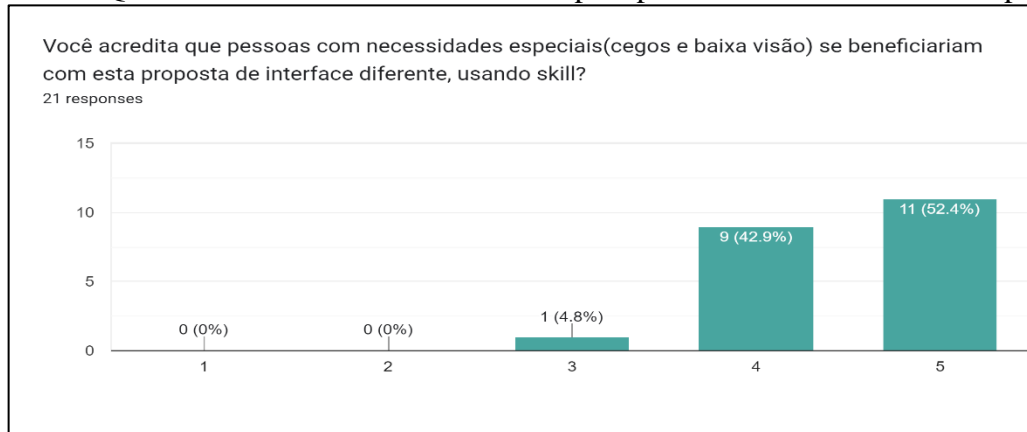
Fonte: elaborado pelo autor.

Figura 66 - Questionário - resultado conclusão sem auxílio externo



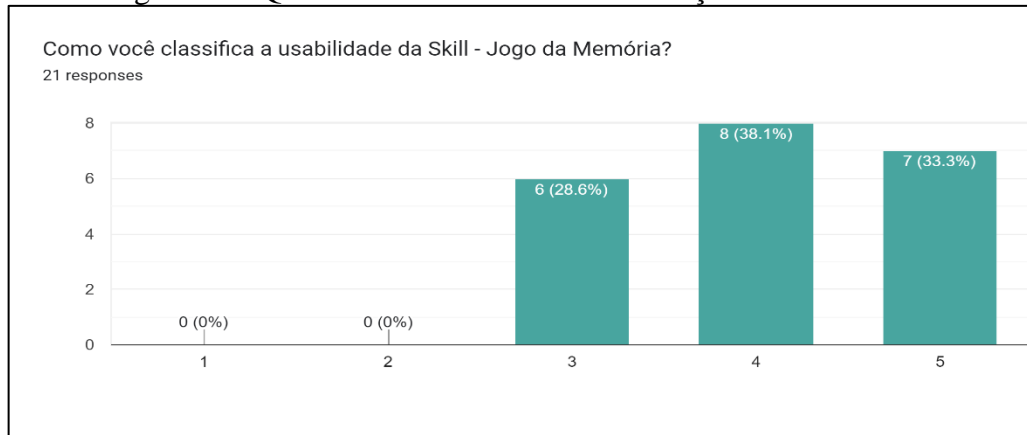
Fonte: elaborado pelo autor.

Figura 67 - Questionário - resultado usabilidade para pessoas com necessidades especiais



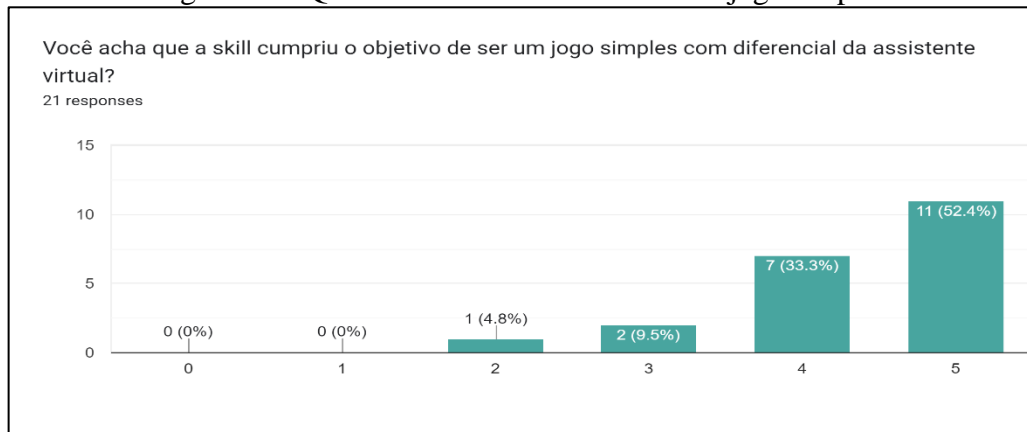
Fonte: elaborado pelo autor.

Figura 68 - Questionário - resultado classificação da usabilidade



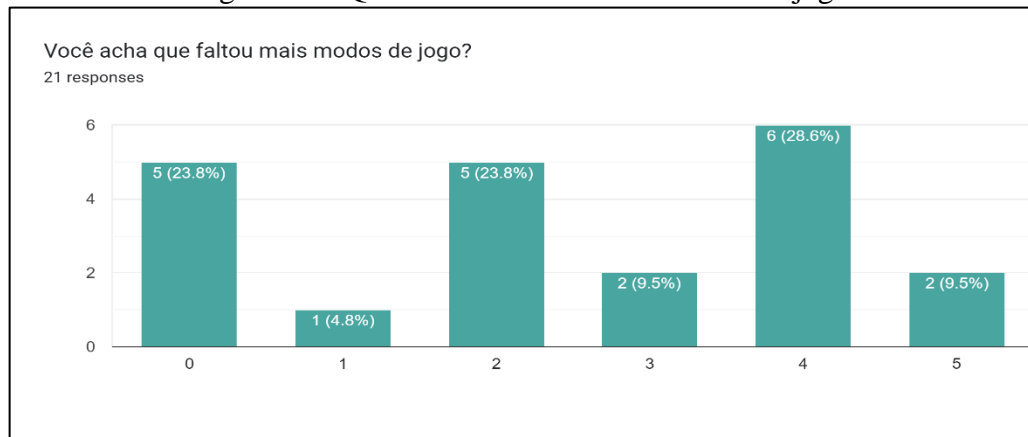
Fonte: elaborado pelo autor.

Figura 69 - Questionário - resultado se é um jogo simples



Fonte: elaborado pelo autor.

Figura 70 - Questionário - resultado modos de jogo



Fonte: elaborado pelo autor.

Figura 71 - Questionário - resultado críticas, comentários ou sugestões

Você possui algum comentário geral, crítica ou sugestão?

9 responses

Consegui concluir todos os modos de jogo, apenas o ingles nao, pois a alexa nao reconheceu o meu ingles. Achei interessante a proposta do jogo, muito facil de aprender a jogar.

A dificuldade está muito elevada, no sentido que o jogo começa muito rápido, no jogo padrão de genius a quantidade de cores para memorizar aumenta aos poucos

Gostei muito da proposta, acredito que tem muitos cenários que pode ser aplicado.

achei muito legal a ideia

tive muita dificuldade com a parte em inglês, mas me pareceu mais um problema da assistente virtual do que do jogo em si.

Não, gostei da jogabilidade, é divertido, simples e cumpre a sua proposta

Quando eu falei a dificuldade e o modo de jogo, começou instantaneamente a mostrar a sequencia de cores, talvez dar um tempo

Fonte: elaborado pelo autor.

Figura 72 - Questionário - resultado erro ou melhorias

Você encontrou algum erro ou melhoria que pode ser feito no aplicativo?

10 responses

Melhoria seria colocar modos muito mais difíceis, para ser um jogo ainda mais desafiador e prender a atenção por mais tempo. E o erro na minha experiência foi com a parte em inglês, que infelizmente não consegui testar, fiz mais de 10 tentativas de maneiras diferentes de falar, e ela não conseguiu reconhecer.

Não consegui fazer o modo em inglês, tentei diversas vezes mas não teve jeito.

Modo inglês não funciona.

Quando você solicita o início, a assistente ainda está explicando/falando e a sequência de cores já começa a aparecer. Isso me deixou bem confuso e errei umas 3 vezes até perceber que isso acontecia, pois estava olhando para onde o som sai ao invés do aparelho memory. Acredito que ela deva ter respostas mesmo para quando não entende o comando.

O modo inglês está falhando

A Alexa só responde ou entende em inglês quando fechar o jogo da memória e abrir novamente, em inglês.

parte em inglês não está respondendo muito bem, dando outras informações ou só ignorando mesmo

Fonte: elaborado pelo autor.

Figura 73 - Questionário - resultado erros ou melhorias (continuação)

Você encontrou algum erro ou melhoria que pode ser feito no aplicativo?

10 responses

Modo inglês não funciona.

Quando você solicita o início, a assistente ainda está explicando/falando e a sequência de cores já começa a aparecer. Isso me deixou bem confuso e errei umas 3 vezes até perceber que isso acontecia, pois estava olhando para onde o som sai ao invés do aparelho memory. Acredito que ela deva ter respostas mesmo para quando não entende o comando.

O modo inglês está falhando

A Alexa só responde ou entende em inglês quando fechar o jogo da memória e abrir novamente, em inglês.

parte em inglês não está respondendo muito bem, dando outras informações ou só ignorando mesmo

Não

A parte em inglês não está funcionando com a alexa.

Em inglês eu não consegui

Fonte: elaborado pelo autor.