

UNIVERSIDADE FEEVALE

GUSTAVO MAIER KNEWITZ

**PROTÓTIPO DE SKILL PARA ASSISTENTE VIRTUAL ALEXA COMO INCENTIVO  
AO TRATAMENTO DE DEPRESSÃO**

Novo Hamburgo

2020

GUSTAVO MAIER KNEWITZ

**PROTÓTIPO DE SKILL PARA ASSISTENTE VIRTUAL ALEXA COMO INCENTIVO  
AO TRATAMENTO DE DEPRESSÃO**

Trabalho de Conclusão de Curso apresentado  
como requisito parcial à obtenção do grau de  
Bacharel em Ciência da Computação pela  
Universidade Feevale

Orientador: Prof. Sandra Teresinha Miorelli

Novo Hamburgo

2020

## **AGRADECIMENTOS**

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

Aos meus pais, Carlos e Cristina, por sempre serem muito compreensivos e paciosos ao longo dessa jornada.

Ao meu irmão Guilherme, por todo o auxílio durante o desenvolvimento deste trabalho, sempre debatendo ideias e colocando-se disponível para o que fosse preciso.

Aos meus amigos Carlos e Thaís, por todo o suporte dado, com certeza foi essencial para que eu chegasse até aqui.

Muito obrigado!

## RESUMO

A depressão é uma doença que atinge grande parte da população mundial. É uma doença psicológica que vem aumentando de forma expressiva nos últimos anos. Ela é caracterizada por um sentimento muito profundo de tristeza, perda de interesse e desesperança. Desta forma, impacta substancialmente na realização de atividades cotidianas e na qualidade de vida dos portadores e até mesmo dos familiares. É considerada uma das doenças mais incapacitantes e em casos mais graves pode levar ao suicídio. O tratamento normalmente é feito através do acompanhamento de um psicólogo ou psiquiatra, podendo ser assistido, em alguns casos, do uso de medicamentos. Uma das maiores barreiras contra a doença ainda é a falta de tratamento adequado às pessoas afetadas, cerca de metade dos casos relatados não praticam nenhum tipo de tratamento. Pensando nisso, este trabalho teve como objetivo o desenvolvimento de um protótipo de skill da assistente virtual da Amazon, a Alexa, para incentivar o tratamento da depressão através do uso de técnicas comportamentais, psicológicas e diálogos instrutivos e interventivos, de forma a ter um acompanhamento mais regular e eficiente. Foram identificadas vertentes da psicologia que estavam mais alinhadas com o objetivo do trabalho e com base nisso foi desenvolvido o protótipo, também foram aplicadas diversas técnicas de VUI a fim de melhorar a experiência do usuário e trazer um bom engajamento à aplicação. O presente trabalho ainda obteve uma boa aceitação por parte dos profissionais da área da psicologia que validaram o protótipo desenvolvido.

Palavras-chave: Depressão. Assistente virtual. Criação de skill. Alexa. VUI.

## **ABSTRACT**

Depression is a disease that affects a large part of the world population. It is a psychological disease that has been increasing significantly in recent years. It is characterized by a very deep feeling of sadness, loss of interest and hopelessness. In this way, it substantially impacts the performance of daily activities and the quality of life of patients and even family members. It is considered one of the most disabling diseases and in more severe cases it can lead to suicide. Treatment is usually done through the monitoring of a psychologist or psychiatrist, and can be assisted, in some cases, with the use of medications. One of the biggest barriers against the disease is the lack of adequate treatment for the affected people, about half of the reported cases do not practice any type of treatment. With this in mind, this work aimed to develop a skill prototype for Amazon's virtual assistant, Alexa, to encourage the treatment of depression through the use of behavioral, psychological techniques and instructive and interventional dialogues, in order to have a follow-up more regular and efficient. Strands of psychology were identified that were more aligned with the objective of the work and based on this the prototype was developed, several VUI techniques were also applied in order to improve the user experience and bring a good engagement to the application. The present work still received a good acceptance on the part of the psychology professionals who validated the developed prototype.

**Keywords:** Depression. Virtual assistant. Skill creation. Alexa. VUI.

## LISTA DE FIGURAS

Figura 1 - Critérios de diagnóstico para depressão conforme DSM-5.....	15
Figura 2 - Fluxo do funcionamento da Alexa .....	21
Figura 3 - Composição de uma interação Alexa.....	23
Figura 4 - Comparativo entre GUI e VUI .....	26
Figura 5 - Diálogo simulado de variedade de resposta .....	28
Figura 6 - Exemplo de divulgação progressiva.....	32
Figura 7 - Exemplo de fluidez de conversa .....	33
Figura 8 - Emoções da Cortana .....	34
Figura 9 - Exemplo de falta de variedade.....	35
Figura 10 - Exemplo de variedade aplicada no diálogo.....	35
Figura 11 - Parte do fluxo de funcionamento do protótipo.....	38
Figura 12 - Conteúdo de pensamentos automáticos.....	40
Figura 13 - Sugestão de acompanhamento .....	42
Figura 14 - Criação de skill.....	44
Figura 15 - Definições criação de skill.....	45
Figura 16 - Seleção de template de skill .....	46
Figura 17 - Console de desenvolvimento .....	46
Figura 18 - Nome de invocação da skill .....	47
Figura 19 - Enunciados com slots .....	48
Figura 20 - Handler de inicialização .....	49
Figura 21 - Construtor de skill .....	51
Figura 22 - Handler de respostas parte 1 .....	52
Figura 23 - Hanlder de respostas parte 2.....	53
Figura 24 - Handler de sorteio de conteúdos .....	54
Figura 25 - Hanlder de conteúdo específico.....	55
Figura 26 - Handler de sugestão de acompanhamento .....	56
Figura 27 - Tags SSML aplicadas .....	57
Figura 28 - Questões do formulário de validação.....	59
Figura 29 - Resultados da validação da aplicação .....	60
Figura 30 - Resultados da validação dos conteúdos.....	60

## LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
ASK	<i>Alexa Skills Kit</i>
ASR	<i>Automatic Speech Recognition</i>
CID-10	Classificação Internacional de Doenças 10ª Revisão
DSM-5	Manual de Diagnóstico e Estatístico de Transtornos Mentais 5ª Edição
GUI	<i>Graphic User Interface</i>
IOT	<i>Internet Of Things</i>
IPA	<i>International Phonetic Alphabet</i>
NLU	<i>Natural Language Understanding</i>
RPD	Registro de Pensamentos Disfuncionais
SSML	<i>Speech Synthesis Markup Language</i>
TCC	Terapias Cognitivo Comportamentais
UI	<i>User Interface</i>
VUI	<i>Voice User Interface</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>10</b>
1.1 OBJETIVOS .....	12
<b>1.1.1 Objetivo geral</b> .....	<b>12</b>
<b>1.1.2 Objetivos específicos</b> .....	<b>12</b>
1.2 ESTRUTURA DO TRABALHO .....	13
<b>2 DEPRESSÃO</b> .....	<b>14</b>
2.1 DIAGNÓSTICO .....	14
2.2 TRATAMENTO .....	15
<b>2.2.1 Terapias cognitivo-comportamentais</b> .....	<b>16</b>
<b>2.2.1.1 Psicoeducação</b> .....	<b>17</b>
<b>2.2.1.2 Terapia de autocontrole</b> .....	<b>17</b>
<b>2.2.1.3 Terapia de resolução de problemas</b> .....	<b>18</b>
<b>3 ALEXA</b> .....	<b>19</b>
3.1 PALAVRA DE ATIVAÇÃO .....	19
3.2 AMAZON LEX .....	20
3.3 FUNÇÃO LAMBDA .....	20
3.4 INTENÇÃO .....	21
3.5 ENUNCIADOS .....	22
3.6 SLOTS .....	22
3.6 CERTIFICAÇÃO E PUBLICAÇÃO NA STORE .....	24
<b>4 INTERFACE DE VOZ AO USUÁRIO</b> .....	<b>25</b>
4.1 TIPOS UI .....	25
4.1 VANTAGENS VUI .....	25
4.2 FLEXIBILIDADE DE ENTRADAS .....	27
4.3 PRINCÍPIO COOPERATIVO .....	28
<b>4.3.1 Quantidade</b> .....	<b>29</b>
<b>4.3.2 Qualidade</b> .....	<b>30</b>



<b>4.3.3 Relação</b> .....	<b>31</b>
<b>4.3.4 Maneira</b> .....	<b>31</b>
4.4 DIVULGAÇÃO PROGRESSIVA.....	31
4.5 PERSONALIDADE .....	32
4.6 CONTINUIDADE E FLUIDEZ.....	33
4.7 MULTIMODAL, VOZ E TELA .....	34
4.8 VARIEDADE .....	34
<b>5 DESENVOLVIMENTO</b> .....	<b>36</b>
5.1 DIÁLOGOS / TEMAS .....	39
5.2 APLICAÇÃO DA VUI.....	42
5.3 CURSO AMAZON .....	43
5.4 DESENVOLVIMENTO DO PROTÓTIPO .....	44
<b>5.4.1 Função Lambda</b> .....	<b>49</b>
<b>5.4.2 SSML</b> .....	<b>57</b>
<b>5.4.3 Dificuldades</b> .....	<b>57</b>
5.5 VALIDAÇÃO DO PROTÓTIPO .....	58
<b>6 CONCLUSÃO</b> .....	<b>62</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>64</b>
<b>APÊNDICE A – FLUXO COMPLETO DO PROTÓTIPO</b> .....	<b>67</b>
<b>APÊNDICE B – QUESTIONÁRIO DE VALIDAÇÃO</b> .....	<b>68</b>
<b>APÊNDICE C – CÓDIGO DESENVOLVIDO PARA A APLICAÇÃO</b> .....	<b>78</b>

## 1 INTRODUÇÃO

A depressão é uma doença comum, que afeta mais de 264 milhões de pessoas no mundo. Dados da Organização Mundial da Saúde estimam que no Brasil, cerca de 5,8% da população sofre da doença, o que nos faz liderar o ranking de maior índice de depressão nos países da América Latina e a ocupar a segunda posição no ranking mundial, ficando atrás apenas dos Estados Unidos por 0,1%. A média mundial deste ranking é de 4,4%, valor significativamente menor que os apresentados pelo Brasil, o que evidencia a necessidade de atenção a doença no país. Uma das maiores barreiras no combate a depressão ainda é a falta de tratamento adequado às pessoas afetadas, estudos apontam que cerca de metade dos casos relatados não praticam nenhum tipo de tratamento (OPAS, 2017).

Além disso, ela é a segunda maior causa de mortalidade de jovens na faixa de 15 a 29 anos, situando-se logo após os acidentes de trânsito (WHO, 2017). Em 2017 foi feita uma pesquisa pela Unicef com o intuito de verificar o impacto da tecnologia na vida dos jovens, constatou-se que a faixa etária mais conectada e ligada a tecnologia são jovens entre 15 a 24 anos (UNICEF, 2017). Idades que se assemelham a maior incidência de suicídio. Uma motivação para este trabalho é a crença de que tratamentos médicos em conjunto com ferramentas tecnológicas podem ajudar tanto o paciente quanto o profissional que o acompanha.

Em um estudo feito pela iProspect em 2018, foi registrado que a América Latina realizava 46% mais buscas por voz através de assistentes virtuais que o resto do mundo, dando destaque ao uso das gerações mais jovens. Além desta tecnologia já ter passado a margem de 50% de penetração de mercado, segundo a ComScore a previsão é que em 2020, 50% das buscas serão feitas através de voz (IPROSPECT, 2019). Este estudo foi realizado com mais de 4.000 entrevistados via questionário online. Onde destes, apenas 27% possuem assistentes de voz domésticos, a maioria utiliza as assistentes de smartphone, contudo 70% dos entrevistados que não possuem assistentes domésticas, afirmaram interesse em adquirir uma.

Ao longo dos últimos anos a tecnologia de voz está se infiltrando cada vez mais em nossa vida cotidiana, podemos perceber que não há mais nada de incomum em pessoas conversando com seus alto-falantes, carros, computadores e telefones. A utilização de voz se dá nos mais variados momentos, entre eles os mais relatados são quando a pessoa está em movimento, dirigindo, cozinhando, assistindo tv ou na cama.

De fato, esta tecnologia está mudando a forma com que interagimos com nossos dispositivos.

Nos dias atuais, as assistentes virtuais vêm ganhando crescente espaço e atenção no contexto de tecnologia (SONDA, 2017), no entanto, ainda são popularmente usadas para tarefas simples como gerenciamento de alarmes e datas, pesquisas a internet e pequenas automações residenciais. Com a forte presença da Internet das Coisas (IoT) atualmente, o número de aparelhos conectados por usuário tem aumentado cada vez mais e com isso surge uma necessidade de praticidade ao gerenciar esses dispositivos. Neste contexto, originam-se as assistentes virtuais, que podem auxiliar nas mais diversas tarefas, como por exemplo, na automação residencial, realizando tarefas como apagar as luzes ou fazer um pedido de comida a um restaurante apenas com um comando de voz. A praticidade que as assistentes virtuais trazem é muito grande, pois pode-se estar com as mãos livres, longe do dispositivo alvo e não há a necessidade de acessar menus e formulários para realizar o que desejamos. Ou seja, é uma forte tendência para os próximos anos (SONDA, 2017).

Entre as assistentes virtuais mais populares como a Siri da Apple e o Google Home do Google, existe a Alexa da Amazon que possui cerca de 28% deste mercado, fazendo com que ela seja a líder do setor (KINSELLA, 2020). Além de seus próprios alto falantes inteligentes, como são chamados, a empresa disponibiliza a própria assistente virtual para que outros fabricantes de hardware possam integrar a Alexa em seus produtos (DESMAZIERES, 2019). No início de 2019 foi anunciada a chegada da assistente virtual Alexa na versão português do Brasil, junto da possibilidade de desenvolver suas próprias habilidades e disponibilizá-las para publicação na *Alexa Skills Store* (DESMAZIERES, 2019), que atualmente conta com mais de 350 *skills* disponíveis para serem utilizadas no idioma português do Brasil. Em uma busca à *Alexa Skill Store* não foram encontradas *skills* com o intuito de auxiliar pessoas com depressão, apenas para auxiliar na saúde, acompanhar a gravidez e comunicar parentes sobre algum acidente ou algo do gênero. São alguns exemplos de *skills*: *Everyday Health*, *Mayo Clinic* e *Ask My Buddy*.

Recentemente a Amazon lançou algumas opções de entonações para a Alexa, com o intuito de trazer mais realidade aos seus diálogos. São 6 entonações para simular diferentes estágios de emoção e com isso aproximar os usuários de uma experiência cada vez mais realista. Por enquanto essa característica só está

disponível nos EUA mas já mostra que avanços na área das assistentes virtuais estão chegando e conquistando progressivamente mais usuários (GAO, 2019). Entende-se que este e outros aprimoramentos que estão por vir podem fortalecer o vínculo do paciente com a *skill* de forma a deixar um diálogo menos robotizado e mais orgânico.

Portanto, este trabalho propôs o desenvolvimento de um protótipo de *skill* da assistente virtual Alexa para incentivar o tratamento a depressão. Ao iniciar a habilidade há uma identificação se o usuário já teve acompanhamento psicológico anteriormente ou não, para que assim, possa sugerir seu retorno às consultas ou indicar o início do acompanhamento psicológico. A proposta do protótipo não é substituir o trabalho do profissional, e sim, contribuir para com ele. A aplicação se dá em forma de diálogo simulado, no qual são explicados alguns conceitos, técnicas comportamentais e psicológicas, junto de diálogos instrutivos e interventivos para obter assim um reforço ou embasamento dos conceitos utilizados na terapia.

A aplicação não é para uso vitalício, apenas para um período, pois as questões abordadas possuem uma diversidade, mas chegam a um fim, após isso deve-se procurar um profissional que possa dar continuidade ao trabalho.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

Desenvolver um protótipo de *skill* para a assistente virtual Alexa, na língua portuguesa, para incentivar o tratamento de depressão psicológica

### 1.1.2 Objetivos específicos

- Identificar abordagens da psicologia que a *skill* possa atuar em prol do tratamento de depressão;
- Realizar o estudo bibliográfico quanto a ferramentas necessárias para o desenvolvimento de *skills*;
- Montar uma base de conhecimento que possa ser utilizada para outras áreas de conhecimento;
- Desenvolver um protótipo de *skill* que empregue as abordagens levantadas;

- Validar a ferramenta desenvolvida;

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em 6 capítulos. O primeiro capítulo apresenta uma introdução sobre a proposta do trabalho. Após isso, temos um aprofundamento a respeito de depressão no capítulo 2, abordando a parte mais técnica da doença e formas reconhecidas de tratamento. No capítulo 3, são apresentadas informações sobre a Alexa, desde seu funcionamento até a composição de uma skill. Em seguida, no capítulo 4, temos uma explicação sobre os conceitos e boas práticas de Interface de Voz ao Usuário, de forma a elucidar como se dá um bom desenvolvimento de aplicação de voz. Posteriormente no 5 capítulo, temos o desenvolvimento do protótipo de skill propriamente dito e por fim, as conclusões sobre a pesquisa são expostas no capítulo 6.

## 2 DEPRESSÃO

A depressão é um distúrbio mental comum que se apresenta como uma profunda tristeza ou perda de interesse em atividades que antes eram agradáveis ou prazerosas (WHO, 2020). Além disso, a depressão muitas vezes vem acompanhada de sintomas de ansiedade. Os efeitos da doença podem ser duradouros ou recorrentes e por isso impactam substancialmente na realização de atividades cotidianas dos portadores, em casos mais graves pode levar ao suicídio. Segundo a WHO (2020), é considerada uma das doenças mais incapacitantes e uma grande contribuidora para a manifestação e surgimento de outras doenças.

### 2.1 DIAGNÓSTICO

O diagnóstico da depressão normalmente é feito através da escuta atenta às queixas do paciente de forma a identificar fatores que estejam sendo negligenciados ou não verbalizados (BARCELLOS et al., 2017). Como a depressão é uma doença que pode ter inúmeros desencadeadores e formas de manifestação, são utilizadas classificações diagnósticas como o CID-10 (Classificação Internacional de Doenças e Problemas Relacionados à Saúde) e o DSM-5 (Manual Diagnóstico e Estatístico de Transtornos Mentais) para efetuar um diagnóstico mais preciso. Eles constituem a melhor informação disponível sobre os transtornos mentais e demais doenças, reunindo informações clínicas como duração, persistência, abrangência e desproporção em relação a um fator desencadeante. O DSM-5 é a publicação mais recente disponível no momento, nesta versão constam nove critérios para classificar a depressão, onde o paciente deve conter pelo menos cinco para ser diagnosticado. Para garantir maior assertividade é necessário que estes sintomas estejam presentes a pelo menos duas semanas e que obrigatoriamente um deles seja humor deprimido ou perda de interesse/prazer (BARCELLOS et al., 2017). Na figura 1 encontra-se a tabela de classificação do DSM-5 com os critérios de classificação.

**Figura 1 - Critérios de diagnóstico para depressão conforme DSM-5**

<p>Presença de pelo menos cinco entre os nove critérios.</p> <p>Sintomas devem persistir por pelo menos duas semanas e um deles deve ser obrigatoriamente humor deprimido ou perda de interesse/prazer.</p>
<ol style="list-style-type: none"> <li>1. Humor deprimido na maior parte do dia, quase todos os dias, conforme indicado por relato subjetivo (p. ex. sente-se triste, vazio ou sem esperança) ou por observação feita por outra pessoa (p. ex., parece choroso) (<b>Nota:</b> em crianças e adolescentes, pode ser humor irritável).</li> <li>2. Acentuada diminuição de interesse ou prazer em todas ou quase todas as atividades na maior parte do dia, quase todos os dias (conforme indicado por relato subjetivo ou observação).</li> <li>3. Perda ou ganho significativo de peso sem estar fazendo dieta (por exemplo, mudança de mais de 5% do peso corporal em menos de um mês) ou redução ou aumento no apetite quase todos os dias. (<b>Nota:</b> em crianças, considerar o insucesso em obter o peso esperado).</li> <li>4. Insônia ou hipersonia quase diária.</li> <li>5. Agitação ou retardo psicomotor quase todos os dias.</li> <li>6. Fadiga ou perda de energia quase todos os dias.</li> <li>7. Sentimentos de inutilidade ou culpa excessiva ou inapropriada (que podem ser delirantes) quase todos os dias (não meramente autorrecriminação ou culpa por estar doente).</li> <li>8. Capacidade diminuída para pensar ou se concentrar, ou indecisão quase todos os dias (por relato subjetivo ou observação feita por outra pessoa).</li> <li>9. Pensamentos recorrentes de morte (não somente medo de morrer), ideação suicida recorrente sem um plano específico, tentativa de suicídio ou plano específico para cometer suicídio.</li> </ol>

Fonte: Barcellos et al. (2017)

Outros sintomas não contemplados na tabela podem ser encontrados mesmo que indiretamente, algumas possibilidades são: desesperança, pessimismo, baixa autoestima, irritabilidade, retraimento social, esquecimentos, ansiedade, sintomas físicos sem explicação, sintomas paranoides e sintomas obsessivos. O fato de o paciente se encaixar nos critérios da doença não encerra a avaliação, ainda é necessário avaliar o risco de suicídio e a possibilidade de algum sintoma ser decorrente de alguma outra doença ou medicação (BARCELLOS et al., 2017).

## 2.2 TRATAMENTO

O tratamento normalmente é feito através do acompanhamento de um psicólogo ou psiquiatra, podendo ser assistido do uso de medicamentos em casos

mais abrangentes. Dependendo do número e gravidade dos sintomas, um episódio depressivo pode ser classificado como leve, moderado ou grave. No guia de intervenção da WHO, sugere-se os tratamentos de apoio psicossocial básico combinado com medicação antidepressiva ou psicoterapia, como terapia cognitivo comportamental, psicoterapia interpessoal ou tratamento para solução de problemas (MARCUS et al., 2012). Medicamentos como forma de tratamento podem ser bem eficazes, porém, não como uma linha de frente no combate à doença, seu uso é mais recomendado a casos moderados e graves. Como complemento aos cuidados dos profissionais da área, é sugerido o uso da autoajuda como uma importante abordagem para auxiliar no tratamento a depressão, pois vários estudos demonstram que livros ou programas de autoajuda baseados na internet ajudam a tratar ou reduzir a depressão (MARCUS et al., 2012).

Entretanto uma das maiores barreiras contra a doença ainda é a falta de tratamento adequado às pessoas afetadas, cerca de metade dos casos relatados não praticam nenhum tipo de tratamento (WHO, 2020).

### **2.2.1 Terapias cognitivo-comportamentais**

As terapias cognitivo comportamentais (TCC), são terapias onde há a modificação de comportamento e crenças. Elas abrangem tratamentos que visam mudar o comportamento alterando pensamentos, interpretações, pressupostos e estratégias de resposta (DOBSON, 2006). As terapias cognitivo comportamentais possuem 3 características básicas: A crença afeta o comportamento; A crença pode ser monitorada e alterada; A mudança comportamental desejada pode ser efetuada por meio da mudança cognitiva.

Diversas abordagens terapêuticas incorporam a terapia cognitivo comportamental, onde se pressupõe que ocorram processos internos chamados de “pensamentos” ou “crenças” e que estes podem gerar a mudança cognitiva (DOBSON, 2006). Essas abordagens pressupõem que a mudança comportamental não necessita envolver mecanismos cognitivos muito complexos ou elaborados, mas sim depender muito mais da ação de mudança de comportamento vinda do paciente. Os resultados das TCCs variam de paciente para paciente, porém normalmente são usados dois parâmetros para medir essa mudança, as crenças e o comportamento. Até certo ponto as mudanças emocionais e fisiológicas também podem ser



indicadores, principalmente quando fazem parte do problema focado na terapia. É importante destacar que existem diferentes classes de terapias e que elas são voltadas para diferentes graus de mudança cognitiva (DOBSON, 2006).

#### 2.2.1.1 Psicoeducação

Uma das abordagens utilizadas dentro da TCC é a psicoeducação, que tem o importante papel de educar o indivíduo sobre sua doença, consequências de um comportamento, construção de crenças, valores, sentimentos e como estes repercutem em sua vida e na dos outros (NOGUEIRA et al., 2017). Nogueira afirma que diversas pesquisas aplicaram diferentes técnicas de terapia cognitivo comportamental para o tratamento de doenças como depressão, ansiedade, estresse pós-traumático e déficit de atenção. Dentre estas técnicas, a que se destacou com os melhores resultados foi a psicoeducação. Para compreender esta abordagem é necessário entender melhor alguns conceitos como: a relação entre pensamentos, sentimentos e comportamentos, bem como os erros causados por pensamentos automáticos, que muitas vezes presentes em transtornos mentais. Além disso, a psicoeducação favorece uma melhor adesão ao tratamento farmacológico, auxilia a promoção de hábitos saudáveis e a qualidade no estilo de vida, bem como colabora para o desenvolvimento da autoestima e bem-estar do indivíduo (NOGUEIRA et al., 2017).

#### 2.2.1.2 Terapia de autocontrole

As abordagens da terapia de autocontrole abrangem diversas técnicas, modelos e estratégias. Os modelos podem ser abstrações de processos naturais ou podem ser princípios para controlar o próprio comportamento. Essa terapia pressupõe que naturalmente todas as pessoas consigam se comportar como duas personalidades, onde uma é controlada e interage com situações e consequências internas e externas. A outra seria uma personalidade controladora, que pode manipular as situações e consequências internas e externas, para atingir algum objetivo (DOBSON, 2006). A intervenção é feita na personalidade controladora, de modo a ensinar novos processos controladores derivados de conceitos psicológicos e princípios de modificação comportamental. Os procedimentos de terapia de

autocontrole não devem ser caracterizados como a aplicação de princípios de aprendizagem e sim, como a forma de ensino desses princípios ao indivíduo, para que ele possa aplicá-los a si mesmo. O foco desta terapia é ensinar estratégias de forma generalizada sobre a vida, para que a pessoa possa aplicar em situações semelhantes quando surgirem. Acredita-se que pessoas com depressão atribuam as causas de eventos negativos a fatores internos e atribuam os eventos positivos a causas externas, essa forma de conduta traz culpa e desamparo as pessoas que assim pensam, por conta disso se sentem constantemente responsáveis por eventos negativos e incapazes de produzir coisas positivas (DOBSON, 2006).

### 2.2.1.3 Terapia de resolução de problemas

É um processo onde a pessoa tenta identificar ou definir soluções efetivas a respeito de problemas da vida cotidiana (DOBSON, 2006). Neste contexto é defendida a importância da distinção dos conceitos de “resolução de problemas” e “implementação de soluções”, esses processos são diferentes e por isso exigem diferentes conjuntos de habilidades. A resolução de problemas refere-se a encontrar ou definir soluções para problemas específicos, enquanto a implementação de soluções seria a execução desta na problemática que está sendo abordada. Presume-se que as habilidades de resolução de problemas sejam mais gerais, ao passo que as de implementação de solução variem de acordo com a situação, tipo de problema e solução identificada. Este método trabalha com base no princípio de que os problemas são naturais da vida e sempre existirão, posteriormente começa a criar e solidificar a habilidade de identificação dos problemas no momento que eles surgem, para que não tomem proporções maiores devido a negligência do mesmo. Como último passo, há a modificação da forma com que os problemas são encarados, para que sejam vistos como desafios e não como catástrofes. Esta técnica também pode contemplar o reconhecimento e correção de crenças irracionais ou outras distorções cognitivas, que normalmente contribuem para a criação de um problema confuso e mal definido (DOBSON, 2006).

### 3 ALEXA

No desenvolvimento de aplicações para voz, existem 3 grandes *players* que disponibilizam seu código de forma aberta para que desenvolvedores autônomos ou empresas possam criar suas próprias aplicações de voz, são eles: Amazon, Google e Microsoft (COATES, 2019). Neste trabalho abordaremos a Alexa por demonstrar maior infiltração de mercado atualmente.

A Alexa é uma assistente virtual desenvolvida pela Amazon e disponibilizada tanto na venda de dispositivos Echos, que são os alto falantes inteligentes, quanto embarcada em diversos dispositivos de terceiros (DESMAZIERES, 2019). Ela possui interações próprias pré-estabelecidas com o usuário, porém com uma quantidade limitada, qualquer funcionalidade que exceda essas interações é abordada através de *skills*, que são habilidades da assistente virtual, como se fossem aplicativos de celular (COATES, 2019). Desta mesma maneira as *skills* ficam disponíveis na *Alexa Skill Store* para que o usuário possa buscar e utilizar novas habilidades de seu interesse, lá encontram-se todas as *skills* já desenvolvidas para a assistente virtual de cada país.

No que diz respeito ao desenvolvimento de uma nova *skill*, a Amazon disponibiliza o *Alexa Skills Kit* (ASK) que contém diversas ferramentas e APIs para auxiliar neste processo de criação de *skill*. Dentre estas ferramentas encontram-se editores de código online, servidores para hospedar a aplicação, gerenciador de habilidades e um ambiente de testes que permite a execução de forma web ou em seu próprio dispositivo Alexa (COATES, 2019). Em seguida é abordado o funcionamento do processo realizado pela assistente para atender as solicitações do usuário.

#### 3.1 PALAVRA DE ATIVAÇÃO

Nos dispositivos de voz, é necessária uma palavra de ativação para que a assistente virtual comece a ouvir as solicitações do usuário. Nos dispositivos que possuem Alexa embarcada, o padrão pré-estabelecido desta palavra de ativação é “Alexa”, porém existe a possibilidade de alterar para “Echo” ou “Amazon”, ficando a critério do usuário alterá-la de acordo com sua preferência (COATES, 2019).

O microfone do dispositivo permanece sempre ativo e efetuando uma varredura local no que é dito, a fim de identificar a palavra de ativação e a partir disto “acordar”

o dispositivo para começar a reconhecer o que é dito, impedindo assim que ocorram possíveis confusões e até mesmo a exposição da privacidade de quem possui o aparelho (AMAZON, 2019).

### 3.2 AMAZON LEX

A partir do momento em que a palavra de ativação for usada, a Alexa transmite a fala para um serviço online da Amazon chamado Amazon Lex (COATES, 2019). Esta ferramenta converte a fala em texto para posterior interpretação, trazendo consigo duas funcionalidades implementadas, o Reconhecimento Automático de Fala (ASR) e a Compreensão de Linguagem Natural (NLU) (AMAZON, 2019). Desta forma o desenvolvedor não precisa se preocupar com algo de tamanha complexidade, como fazer a tradução de voz para texto ou a alimentação da base de dados com as pronúncias.

Parte do trabalho executado pela ferramenta é a distinção da voz dos demais sons contidos no ambiente, de forma a reproduzir o que os seres humanos fazem ao prestar atenção em uma conversa (COATES, 2019).

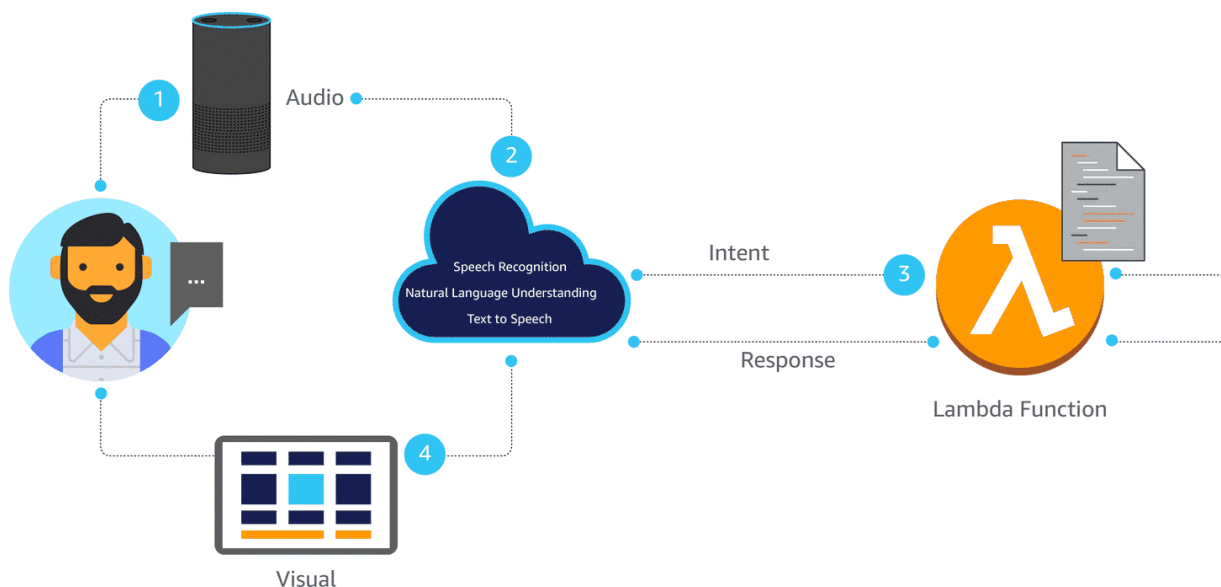
Outro processo desempenhado é assimilar diferentes tons de voz e sotaques, de maneira que o que for dito sempre permaneça intacto, seja uma criança com timbre mais agudo ou um homem mais velho com voz mais grossa (COATES, 2019). O computador precisa interpretar estes dados da mesma forma, mesmo que se apresentem de maneiras diferentes, então ele divide o áudio em pequenos pedaços de forma a virarem fonemas. Deste modo o software compara os fonemas obtidos com os que ele já conhece e depois vai unindo-os até que formem as palavras novamente. Após conseguir montar toda a frase, é enviada uma requisição JSON para a função Lambda que verificará se há alguma intenção que atenda a solicitação do usuário e qual seria ela (COATES, 2019).

### 3.3 FUNÇÃO LAMBDA

É um serviço disponibilizado pela Amazon para executar o seu código na nuvem sem a necessidade de ficar gerenciando servidores. Ele hospeda a aplicação automaticamente com uma infraestrutura de alta disponibilidade, 24 horas por dia, 7 dias da semana. Atualmente o Lambda tem suporte às linguagens Java, Node.js, C#

e Python (AMAZON, 2019). Para os desenvolvedores mais arrojados, pode-se optar em não utilizar o Lambda, porém há a necessidade de procurar e gerenciar um servidor para hospedar e executar as aplicações desenvolvidas.

**Figura 2 - Fluxo do funcionamento da Alexa**



Fonte: Amazon (2020)

A figura 2 apresenta um exemplo de todo o funcionamento de uma interação com a Alexa. O usuário diz a palavra de ativação e faz uma solicitação à Alexa, o dispositivo envia o áudio para o serviço Amazon Lex que converte o áudio em texto e envia via JSON para a função Lambda, que por sua vez analisa a solicitação procurando uma intenção correspondente no código. Após isso, é enviado um retorno via JSON para o Amazon Lex que converterá os dados em áudio novamente e enviará para a Alexa reproduzir ao usuário em forma de resposta (AMAZON, 2019).

### 3.4 INTENÇÃO

As intenções são as ações que o usuário deseja executar na *skill*. Por exemplo, “Alexa, acenda as luzes” (AcenderLuzesIntent) ou “Alexa, apague as luzes” (DesligarLuzesIntent) (COATES, 2019). São duas intenções diferentes que realizam duas ações diferentes, por mais semelhantes que pareçam.

Estas intenções expressam no código qual ação deve ser executada para uma determinada requisição, isso ocorre porque a Alexa não fornece aos desenvolvedores o texto bruto dito pelo usuário. Por isto a Amazon criou uma interface intermediária

para detectar qual ação deve ser executada e suas diferentes formas de chamada (COATES, 2019). Estas intenções são criadas em uma estrutura JSON como um modelo de situação, contendo seus enunciados relacionados a esta ação e seus slots para entrada de dados. O console online da Amazon disponibiliza uma interface amigável para a criação de modelos de intenções, onde a camada JSON fica oculta para o desenvolvedor.

### 3.5 ENUNCIADOS

As intenções são as ações que o usuário deseja solicitar. Porém na comunicação humana costumamos usar diversas formas para solicitar a mesma coisa a uma pessoa, pois essa flexibilidade da fala nos anima e evita a monotonia (AMAZON, 2019). Por isto existem enunciados, que são diferentes amostras de declarações para solicitar uma mesma ação. Desta maneira não há a necessidade de se desenvolver uma intenção para cada forma de expressão do usuário. Normalmente uma intenção pode conter várias expressões de amostra, mas estar ligada diretamente apenas à uma intenção. Abaixo alguns exemplos de declarações para uma mesma tarefa:

“Apague as luzes”

“Desligue as luzes”

“Não quero mais luzes acesas”

“Desligar luzes”

“Apagar luzes”

Com certeza existem outras formas além das citadas acima para solicitar que sejam apagadas as luzes. Quanto mais amostras de declaração a intenção possuir melhor será a experiência do usuário ao utilizar a habilidade.

### 3.6 SLOTS

Os slots são as variáveis contidas na intenção, estão dispostos em alguns lugares das amostras de expressão onde há alguma necessidade de manipulação de dado (COATES, 2019). Utilizando o mesmo exemplo das luzes, provavelmente teríamos um slot a respeito de quais luzes devem ser apagadas:

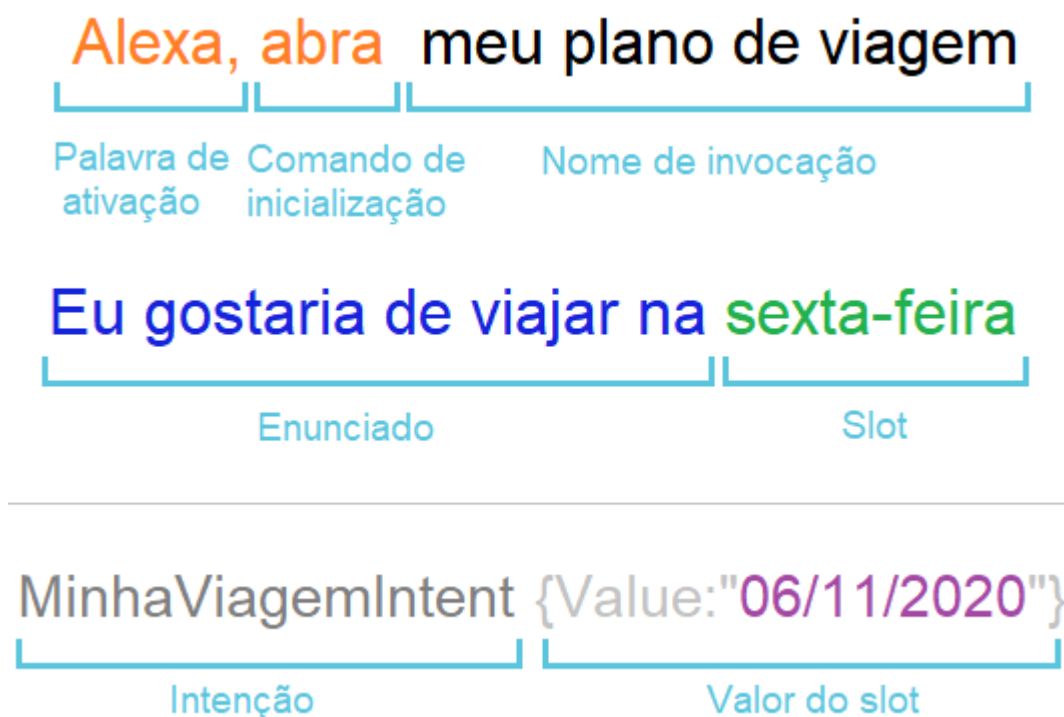
“Apague as luzes da [sala]”

“Apague as luzes da [cozinha]”

“Apague as luzes do [quarto]”

Os slots são compostos por três partes, nome, tipo e amostras de valores aceitos. Os nomes são utilizados para referência nos enunciados e no código da aplicação após receber os valores adequados ao slot. O tipo, diz respeito a qual tipo de informação poderá ser aceita como entrada, como cidades, datas, números, entre outros. Há a possibilidade de utilizar alguns tipos já existentes na Alexa ou criar um personalizado, para isto é necessário definir as amostras de valores aceitos pelo slot, garantindo assim a integridade da aplicação. Tendo em vista que essa limitação de entradas evita de serem preenchidos valores inválidos ou que não condizem com a intenção, acaba-se reduzindo assim a possibilidade de erros na *skill* desenvolvida.

Figura 3 - Composição de uma interação Alexa



Fonte: Elaborado pelo Autor (2020)

Unindo os elementos que compõem uma interação da Alexa obtemos a composição exibida na figura 3. Podemos observar que o usuário pronuncia a palavra de ativação junto com o nome de invocação da *skill*, isto faz com que seja acessado o manipulador de inicialização e a sua respectiva funcionalidade definida em código. Em seguida, a *skill* permanece ativa para ouvir as demandas do usuário, que solicita marcar uma viagem na sexta-feira. Nesta solicitação existem dois elementos

presentes, um enunciado e um *slot*. O enunciado está vinculado a intenção chamada *MinhaViagemIntent*, que executará o agendamento da viagem de acordo com o definido na função *Lambda*. Logo após o enunciado são fornecidos alguns dados de entrada através de um *slot*, que serão utilizados junto com o processamento da solicitação feita. Observa-se ainda que, ao alterar o valor de entrada do *slot* para outro dia da semana, isso não afetará o funcionamento de nenhum outro componente, pois haverá somente a substituição de um valor para o *back-end*.

### 3.6 CERTIFICAÇÃO E PUBLICAÇÃO NA STORE

A *Alexa Skills Store* é a loja de *skills* da Amazon onde ficam disponíveis todas as habilidades já desenvolvidas e que foram aprovadas para a comunidade geral. Ela funciona como uma loja de aplicativos de celular, onde é possível fazer buscas e escolher novas aplicações para a utilização em seu dispositivo. É possível efetuar a publicação da *skill* para um único país ou para uma gama maior de pessoas abrangendo outros países, isto dependerá do idioma padrão utilizado para o desenvolvimento da aplicação e da aprovação da Amazon para tal.

Para que a *skill* esteja disponível à utilização dos usuários em seus dispositivos Alexa, ela deve passar por um processo de certificação feito pela Amazon. Onde são feitos alguns testes para verificar a existência de falhas e posteriormente é feita a certificação de que a *skill* está apta a ser disponibilizada aos usuários finais. Após a publicação na store não é possível fazer alterações na aplicação, caso isso seja necessário é preciso alterar uma cópia e posteriormente passar pelos processos de teste e certificação novamente.



## 4 INTERFACE DE VOZ AO USUÁRIO

As Interfaces de Usuário (UI) são a forma como nós seres humanos interagimos com as máquinas (DASGUPTA, 2018). Isto abrange tudo, desde mouse e teclado até contextos de menu, utilização de cores, etc. À medida que a tecnologia vai mudando, suas interações conseqüentemente acompanham este movimento. Por exemplo, antes da chegada dos celulares *touch screen* não haviam interações como deslizar, beliscar para dar zoom ou rotacionar o dispositivo (DASGUPTA, 2018). Outro exemplo para elucidar o conceito de UI seria o visual de um aplicativo para celular, a forma como os menus estão dispostos e como você navega entre eles é uma forma de interface de usuário. Isto é modelado de acordo com a forma com que o designer de experiência e o arquiteto de software acreditam ser a forma mais prática de utilização do sistema.

### 4.1 TIPOS UI

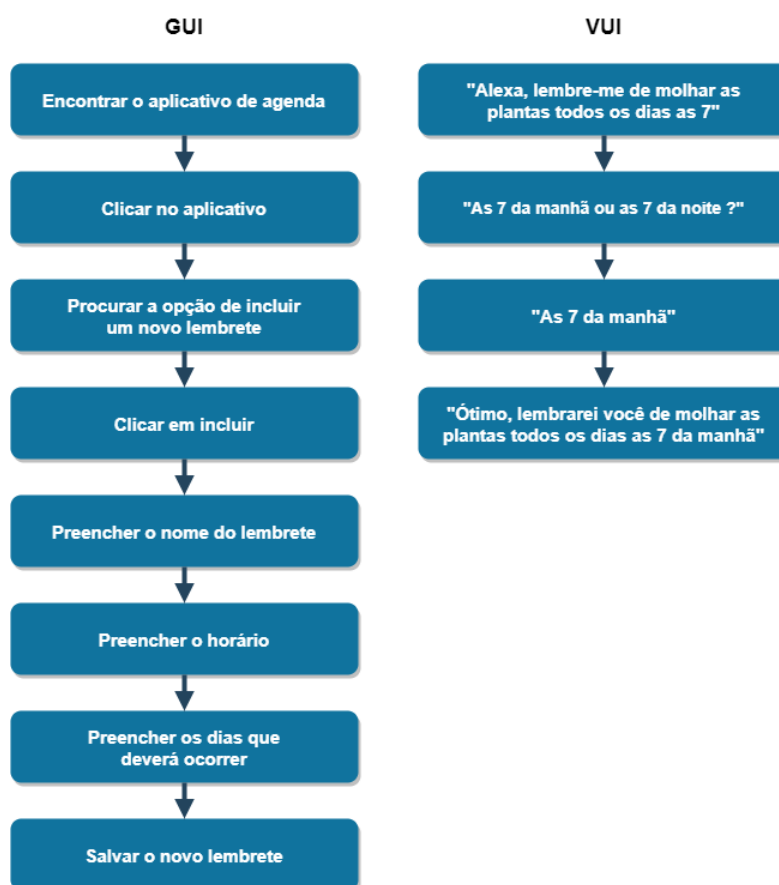
Existem diferentes tipos de UI, dentre elas a mais difundida até hoje é a Interface Gráfica do Usuário (GUI), que compreende a parte visual das aplicações (DASGUPTA, 2018). Mas quando falamos de assistentes virtuais, sua grande maioria não possui tela com elementos gráficos e não tem isto como seu produto principal, e sim, a voz. Desta forma, surgiu o conceito de Interface de Voz do Usuário (VUI), onde permite que os usuários interajam com a tecnologia através do reconhecimento de suas vozes. Por ser uma das principais formas de comunicação dos seres humanos, a fala é uma das interações mais naturais que podemos ter com a tecnologia. Porém, este novo paradigma traz consigo alguns desafios para a interface com o usuário, pois a conversação é algo muito natural para nós. Passamos a vida toda conversando e quando surgem quaisquer problemas neste meio de comunicação é comum que mesmo assim consigamos compreender a fala do outro, devido a capacidade de entender questões de contexto e fazer relacionamentos entre isto (DASGUPTA, 2018).

### 4.1 VANTAGENS VUI

Algumas vantagens que a VUI possui em relação aos demais tipos de interface seria a sua forma mais intuitiva e natural de interação. A GUI é um comportamento que aprendemos, não é algo natural e varia de aplicação para aplicação. Por exemplo, quando os bebês começam a ter contato com telas, após certo tempo aprendem a dominar aquele contexto, mas ao trocar o aplicativo ou sistema operacional, possuem dificuldade, porque a interface gráfica é em sua maioria remodelada de acordo com a aplicação. No contexto de voz, este reaprendizado não ocorre, pois só aprendemos a falar uma vez com as outras pessoas. (DASGUPTA, 2018).

A agilidade e praticidade são outros pontos fortes da VUI, é muito mais rápido definir um lembrete por voz dizendo apenas “lembre-me de molhar as plantas todos os dias as 7” do que a quantidade de cliques que isso requereria com o uso de uma tela. Além disso, possibilita estar com as mãos e olhos ocupados com outra coisa, como cozinhando ou lendo o jornal, tornando assim a vida do usuário muito mais prática (DASGUPTA, 2018). Na figura 4 temos um exemplo dessa praticidade.

**Figura 4 - Comparativo entre GUI e VUI**



Fonte: Elaborado pelo Autor (2020)

## 4.2 FLEXIBILIDADE DE ENTRADAS

De acordo com Dasgupta (2018), como a fala é algo fundamental para a comunicação humana, não podemos desconsiderar a expectativa que teremos nesta comunicação, mesmo que saibamos que estamos falando com uma máquina, esperamos ser compreendidos como em uma conversa entre humanos. A fala é algo complexo e cheio de meandros, porém por ser algo extremamente natural para nós, não percebemos mais esta característica. Ela também é uma das principais formas das pessoas se expressarem e expressarem sua personalidade (DASGUPTA, 2018). Seria frustrante alguém não ser bem compreendido, afinal, ninguém gosta de passar o papel de “tonto” em uma conversa. Desta forma, se uma aplicação em voz não for bem elaborada pode causar uma experiência negativa ao usuário, diminuindo as chances de reutilização (AMAZON, 2019). Por isto existem uma série de pontos que devem ser levados em consideração ao se projetar uma interface de voz para que ela seja bem sucedida.

Ao se projetar GUI normalmente é pensado quais são as informações primordiais e quais são as secundárias, para não sobrecarregar os usuários de informações, porém para a voz esse conceito vai “por água abaixo”. Ninguém quer ouvir um menu de opções como acontece com as árvores telefônicas, deve ser algo pré-estabelecido, como aconteceria normalmente em uma conversa (DASGUPTA, 2018). Um dos pontos mais primordiais na hora de se desenvolver uma interface de voz é que ela seja natural, sendo assim precisamos contar com as inúmeras formas que o usuário poderá responder à um enunciado ou expressar sua intenção, afinal, a fala é muito flexível e por isto não utilizamos sempre as mesmas palavras e ordens para transmitir algo (AMAZON, 2019). Na figura 5, há um exemplo de como existem formas diferentes de responder uma pergunta e como isto pode se tornar uma experiência negativa e até irritante para o usuário se não for levada em consideração.

Figura 5 - Diálogo simulado de variedade de resposta



Fonte: Elaborado pelo Autor (2020)

#### 4.3 PRINCÍPIO COOPERATIVO

O ser humano é um ser sociável e nos socializamos principalmente através da fala. Algo que apesar de completamente natural para nós é extremamente complexo para ensinar a um computador. Existem vários detalhes envolvidos como a troca de turno de conversação e uma espécie de auxílio mútuo para simplificar a conversa, onde existem algumas expressões como “uhmm” e “ãmmm” para sinalizar que ainda

estamos falando e que o seu silêncio não significa a vez do outro falar. Também existem os generosos “mm-hum” e “uh-huh” que demonstram que os ouvintes estão ouvindo e compreendendo o orador (DASGUPTA, 2018). Foi feito um estudo sobre esse tipo de interação humana durante as conversas, onde foi proposto à diversas pessoas que fossem oradores e contassem suas experiências. Durante o discurso do orador os ouvintes foram encarregados de apertarem um botão cada vez que o orador dissesse uma palavra começada com a letra “T”, isto fez com que prestassem menos atenção a conversa e mais a tarefa, ausentando assim esse meio de contribuição durante o diálogo. O resultado do experimento foi que os oradores se sentiram nervosos em certo ponto e começaram a dar ênfase às partes mais importantes de suas falas como uma forma de garantir que estavam sendo compreendidos devido a não haver respostas ou contribuição da outra parte (DASGUPTA, 2018). Isto mostra como o ser humano tende a facilitar os diálogos. Apesar de uma conversa ser algo complexo, é notável como as pessoas inconscientemente cooperam para facilitá-la.

O princípio cooperativo é algo que traz este conceito de cooperação em forma de algumas características que um bom diálogo deve ter para ser efetivo e agradável. Foi mencionado a primeira vez pelo filósofo linguístico Paul Grice ao debater sobre a facilidade que trazemos às conversas para que possam fluir de uma forma melhor (COATES, 2019). Este conceito consiste basicamente em 4 elementos essenciais, sendo eles: Quantidade, Qualidade, Relação e Maneira.

Quantidade – Fale apenas o necessário, mas não muito.

Qualidade – Fale a verdade.

Relação – Fale algo que seja relevante.

Maneira – Fale de forma clara.

#### **4.3.1 Quantidade**

A quantidade é o elemento que pressupõe que o seu parceiro de conversa lhe dará as informações necessárias para compreender ou completar o assunto em debate. Um exemplo seria você perguntar ao seu amigo como foi a apresentação dele na universidade e ele responder com “Quando saí de casa esqueci o *pendrive*, então tive que usar o notebook para apresentar e como ele é mais antigo não tem entrada HDMI, e por isso tive que...” Você não quer reviver toda a situação, minuto a minuto com ele, uma resposta simples como “Foi boa, apesar de ter ocorrido alguns

contratempos” já atenderia a necessidade. Porém se a resposta for muito curta como “Foi boa” também não seria algo muito satisfatório. Portanto, deve haver um consenso desta quantidade de informação. Esse elemento evidencia a diferença da GUI para a VUI, pois a GUI pode conter informações não relacionadas diretamente a tarefa em questão, como anúncios, botões de compartilhamento em rede social, opções de menus e outros (COATES, 2019).

Os usuários ao utilizarem uma GUI podem desviar o olhar da tela, responder outra pessoa ou simplesmente pensar um pouco antes de tomar uma decisão, assim que retomarem a atenção à tela as informações continuarão lá dispostas como antes. Na VUI, entretanto, não ocorre desta maneira, como o tempo é o único recurso presente, o usuário precisa lembrar das informações que já lhe foram passadas e pensar nas próximas. Portanto a VUI assume esta responsabilidade para não sobrecarregar o usuário (COATES, 2019). Mesmo não utilizando um aplicativo de voz, quase todos já passamos por uma experiência de quebra desta regra ao ligar para uma companhia aérea, banco ou atendimento ao cliente. “Se você deseja comprar uma passagem aérea, pressione 1. Se você deseja consultar o saldo de suas milhas, por favor pressione...” Esse nível de detalhamento raramente é utilizado, por isto existe uma maneira muito simples para a resolução deste problema, a decomposição. A decomposição consiste em o desenvolvedor pegar o enunciado criado e ir removendo elementos, sempre verificando se ainda atende ao objetivo inicial. Assim que deixar de atender a demanda, chegou-se ao ponto final, então basta retroceder um passo e manter o resultado. Se o usuário perguntar: “Qual é o tempo para amanhã?” e o aplicativo responder: “O clima para amanhã é de 28 graus e predomínio de sol” ainda não está bom o suficiente para atender a VUI, o mais adequado seria “Amanhã estará 28 graus e com sol”.

Reduzir a quantidade de turnos que a conversa pode levar entre a etapa inicial e a final demonstra mais objetividade e não prende o usuário a deixar de fazer suas tarefas para prestar única e exclusivamente ao assistente de voz (AMAZON, 2019).

#### **4.3.2 Qualidade**

A qualidade diz respeito a evitar a proliferação de informações falsas ou que não se tem o domínio necessário, bem como omissões e insinuações. Por isto esperamos que nosso parceiro de conversação informe essa imperícia ao explicitar

um palpite, por exemplo: “Acho que o Carlos tem uns 40 anos”. A fala segue o princípio da qualidade, mesmo que Carlos tenha 55 anos, pois está claro que não é um dado confiável (COATES, 2019). Na VUI isso se dá tanto nas falas quanto no fato de apresentar alguma opção que não foi tratada ou que possui alguma falha.

### **4.3.3 Relação**

Este elemento requer que o que foi dito tenha relação com o que antecedeu a resposta na conversa. Normalmente nossos diálogos baseiam-se que um assunto está sempre ligado ao anterior. Desta forma, quando descumprido este requisito é evidente um desconforto para uma das partes envolvidas na conversa (COATES, 2019). Conversar com um amigo que está frequentemente trocando de assunto não é uma experiência muito agradável, mesmo que em algumas vezes o orador desvie a conversa para evitar transparecer a falta de domínio em algum assunto. Isto demonstra falta de respeito no nosso entendimento, o mais adequado seria utilizar-se do elemento de qualidade e assumir a falta de domínio.

### **4.3.4 Maneira**

A maneira instrui os conversadores a serem claros, organizados e específicos. Quando as pessoas usam gírias e siglas elas estão descumprindo esta regra, pois estão deixando alguma informação obscura para quem está de fora do grupo (COATES, 2019). Um bom exemplo do não cumprimento desta característica de clareza e organização são as crianças quando tentam contar uma história. Elas normalmente possuem dificuldade neste aspecto e por isso muitas vezes perdemos a paciência ao ouvi-las. “E então eu fui para a escola e ... não ... eu fui ao shopping com minha mãe. Não, meu pai e minha mãe ...”.

## **4.4 DIVULGAÇÃO PROGRESSIVA**

Durante um diálogo, quando há a necessidade de passar opções para escolha do usuário, é recomendado elencá-las de acordo com um nível de relevância para evitar que o usuário “se perca”. Também é importante não fornecer mais que 3 alternativas juntas, pois é muito difícil para o ser humano lembrar o contexto da

conversa e mais de 3 opções de forma simultânea (DASGUPTA, 2018). Caso seja necessário sugira a ele que peça mais opções, criando assim uma divulgação progressiva das informações. Na figura 6 temos um exemplo de divulgação progressiva.

**Figura 6 - Exemplo de divulgação progressiva**



Fonte: Dasgrupta (2008, p. 33)

#### 4.5 PERSONALIDADE

A personalidade é algo inato do ser humano. Ele vê isso como forma de compreender o mundo e se expressar nele, por isso tem a tendência de agregar personalidades as coisas, mesmo que inanimadas, como o design de produtos ou carros, dependendo da forma com que o designer queira passar as qualidades do produto podemos perceber diferentes personalidades nas marcas. E na voz a personalidade está muito mais presente que nas interações comuns. Desta maneira, a atenção a este ponto na hora da modelagem deve ser redobrada (DASGUPTA, 2018).

No caso da Alexa, é importante ser fiel a personalidade básica já desenvolvida por seus criadores. Onde ela é amigável, otimista e gentil, portanto, manter a interface com as mesmas características e algumas variações é uma boa técnica, salientando a cordialidade e respondendo de forma saudosa com certeza este alinhamento estará garantido (AMAZON, 2019).



## 4.6 CONTINUIDADE E FLUIDEZ

Assim como na comunicação humana, na VUI estas características devem constar na interação da assistente virtual com o usuário. Os seres humanos estão acostumados a se auxiliar das mais diversas formas durante o diálogo, por isto é recomendado sempre entregar algo a mais ao usuário durante a resposta, principalmente se for uma resposta simples como “Sim” ou “Não”. É indispensável dar continuidade e fluidez a conversa (DASGUPTA, 2018).

**Figura 7 - Exemplo de fluidez de conversa**



Na figura 7, o Google Assistente poderia ter respondido apenas com a data solicitada, mas entregou algo a mais e tornou a experiência muito mais envolvente, informando quais seriam os times a se enfrentar no retorno da *Premier League* e ofereceu a possibilidade de perguntar mais informações de outros jogos. Além disso, criou uma experiência multimodal (GUI + VUI) ao disponibilizar *cards* com

informações, quando houver a oportunidade de explorar esta opção é sempre enriquecedor ao usuário (DASGUPTA, 2018). *Cards* também são uma forma de agrupar informações relevantes e facilitar o seu consumo.

#### 4.7 MULTIMODAL, VOZ E TELA

Apesar de existir a possibilidade de unir a GUI e a VUI em assistentes virtuais que acompanham tela, se os desenvolvedores não culminarem seus esforços com foco na VUI podem acabar apenas migrando uma interface para outra, fazendo com que a interface de voz se torne infrutífera. Uma boa forma de unir estas duas interfaces é basear-se na personalidade que a voz carrega, de maneira a trazer expressões visuais como se houvesse mesmo outra pessoa na conversa (DASGUPTA, 2018). A intenção não é desassociar-se de uma interface virtual para uma interface humana, mas trazer traços da personalidade e interação que normalmente temos. Podemos observar na figura 8, a Cortana da Microsoft expressa esse conceito muito bem.

Figura 8 - Emoções da Cortana



Fonte: Dasgupta (2019)

#### 4.8 VARIEDADE

Mesmo que os aplicativos de voz desempenhem sempre a mesma tarefa e tenham que solicitar algo, necessariamente não é preciso que isto seja repetitivo.

Existe a possibilidade de entregar isto de diferentes formas (COATES, 2019). Os usuários relatam que uma VUI sem variedade é chata e irritante. Este é outro ponto no qual uma VUI diverge de uma GUI, enquanto uma GUI deve apresentar a mesma estrutura em cada ação, uma VUI se beneficia da variedade (COATES, 2019).

#### Figura 9 - Exemplo de falta de variedade

**Domingo 7:00**  
Eu: Como você está?  
Alexa: Eu estou bem.

**Segunda-feira 22:00**  
Eu: Como você está?  
Alexa: Eu estou bem.

**Quinta-feira 19:00**  
Eu: Como você está?  
Alexa: Eu estou bem.

Fonte: Elaborado pelo Autor (2020)

Percebe-se na figura 9 como isto não soa natural e muito menos divertido. O desenvolvimento deve ser pensando no engajamento e na retenção do usuário, as interfaces de voz devem ser projetadas para fornecer sempre um bom ambiente, seja a primeira vez que seja utilizada ou a centésima vez, por isso este desenvolvimento deve ser projetado para que não se torne algo repetitivo ou monótono (AMAZON, 2019). Na figura 10, temos o mesmo diálogo já com a variedade aplicada nas respostas, tornando muito mais atrativo a conversa.

#### Figura 10 - Exemplo de variedade aplicada no diálogo

**Domingo 7:00**  
Eu: Como você está?  
Alexa: Estou bem, tomando banho de sol agora no meu espaço digital.

**Segunda-feira 22:00**  
Eu: Como você está?  
Alexa: No momento, estou feliz porque você me chamou.

Fonte: Elaborado pelo Autor (2020)

Outro ponto crucial é escrever para os ouvidos, não para os olhos. A forma como nós nos comunicamos verbalmente é muito menos formal do que a maneira como escrevemos, portanto, é pertinente manter uma certa informalidade em sua VUI para não soar estranho, sempre lembrando de manter o tom de uma conversa e tentando humanizá-la o máximo possível (DASGUPTA, 2018).

## 5 DESENVOLVIMENTO

No processo de exploração da área da psicologia foi realizada uma reunião com uma profissional do ramo para um alinhamento e maior apropriação do assunto. A professora Sandra Miorelli fez o contato com a psicóloga Claudia Maria Teixeira Goulart, que é mestre em Psicologia do Desenvolvimento pela UFRGS e atualmente é proprietária e psicóloga clínica da Clínica Mosaico, bem como professora do segmento na Universidade Feevale.

Na reunião realizada, a psicóloga Claudia foi inteirada da proposta de trabalho e deu algumas sugestões a respeito da bibliografia e até mesmo ideias a serem abordadas no protótipo. No que tange a parte bibliográfica foi indicado utilizar o DSM-5 que é onde constam alguns critérios a respeito da classificação de diversos transtornos mentais, entre eles, a depressão. Foi ressaltado que esta ferramenta é amplamente utilizada como um recurso primordial para a pesquisa, formação e prática da psicologia.

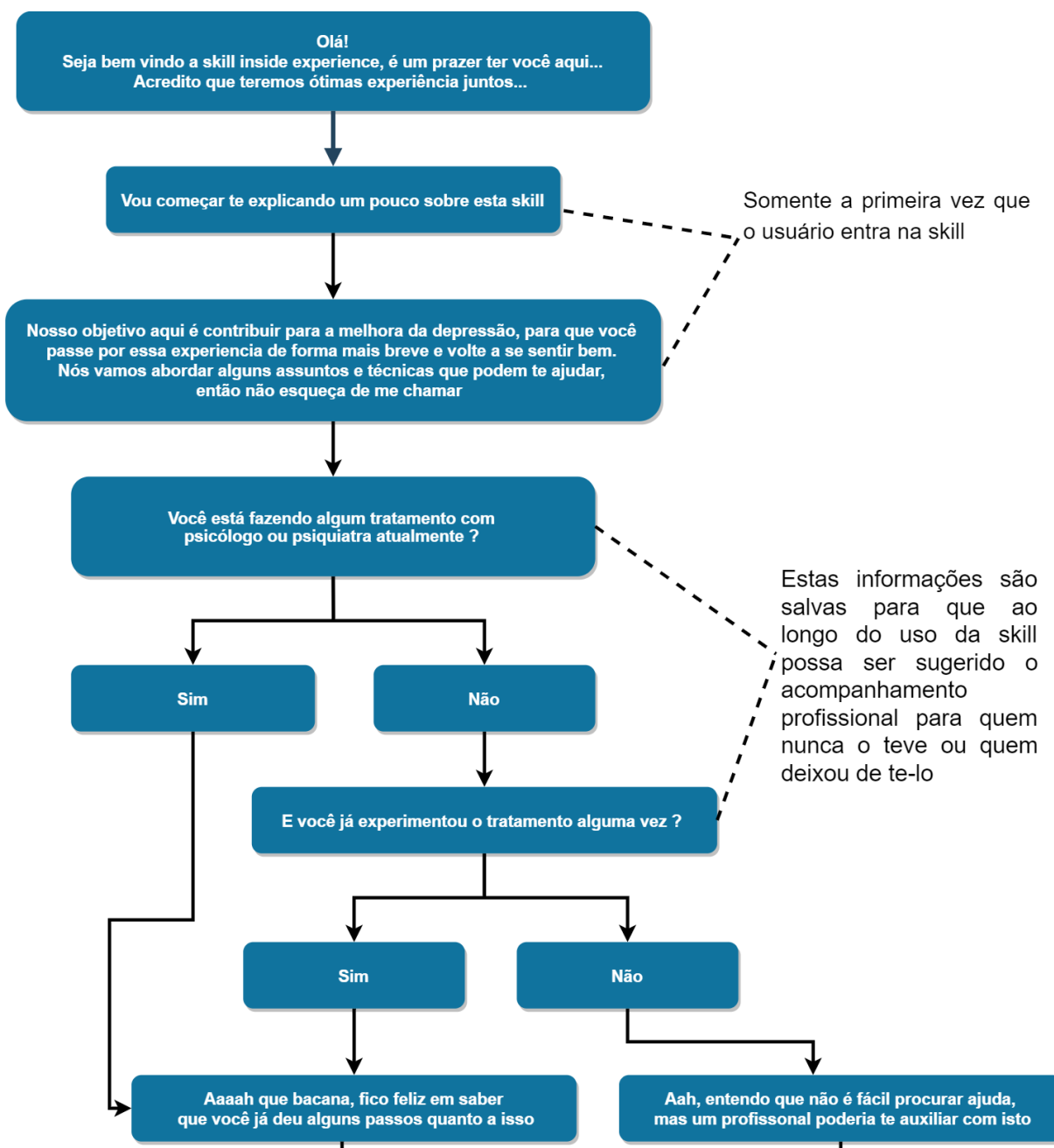
Quanto a parte prática do protótipo, foi sinalizado que a ideia de diálogos instrutivos poderia sim contribuir para melhores resultados e inserção de pacientes no tratamento a depressão. Ainda houve contribuições a respeito de possíveis implementações de ferramentas no protótipo de *skill*. Algumas dessas ideias apontadas pela professora Claudia foram:

- O desenvolvimento de um lembrete de remédios, tendo em vista que muitos pacientes fazem o uso de medicação e a sua não utilização de forma correta pode prejudicar os resultados do tratamento;
- Uma agenda para recordar o paciente de suas consultas, pois o acompanhamento clínico é muito importante, principalmente em casos onde a doença se encontra em um estado mais grave;
- Uma espécie de interação com o paciente onde houvesse a possibilidade de identificar os estados dos pensamentos da pessoa, se andam muito negativos ou não, para que assim pudesse armazenar estes dados em um relatório e disponibilizar ao profissional que for atender a este paciente;
- A última abordagem que surgiu foi a de utilizar-se do DSM-5 para fazer um pré-diagnóstico da pessoa e encaminhar a um profissional. Levando em consideração que seriam algumas perguntas baseadas na classificação da doença no DSM-5. Desta forma o paciente já teria passado por uma triagem e

um profissional poderia o acompanhar de maneira adequada, principalmente porque o diagnóstico não se limita a apenas isso, também há a necessidade de classificar a gravidade e risco de suicídio.

A respeito das sugestões colocadas, foi discutido o grau de complexidade de desenvolver as ferramentas como o lembrete de medicação e de consultas, que mostraram-se muito simples para a proposta de trabalho de conclusão. Também foi questionada a existência de aplicativos com esta funcionalidade em plataformas *mobile*, e que o recurso de voz da assistente virtual não traria nenhum benefício a estas ferramentas, igualando-se às já existentes, por isso foram descartadas estas opções. Os outros dois casos teriam a necessidade de disponibilizar os dados a profissionais capacitados e licenciados, isso poderia trazer problemas de disponibilidade e privacidade dos dados do paciente. Ao passo que a proposta inicial do protótipo visa instruir os usuários a lidar de forma melhor com a doença e indicá-los a retomar ou iniciar o tratamento com um profissional. Encurtando assim, a distância entre pacientes e a ajuda qualificada de um psicólogo ou psiquiatra. Desta maneira foi definido que a proposta inicial seria a desenvolvida, a qual é demonstrado parte de seu funcionamento na figura 11.

Figura 11 - Parte do fluxo de funcionamento do protótipo



Fonte: Elaborado pelo Autor (2020)

Inicialmente há uma mensagem de boas-vindas ao usuário seguida de uma elucidação do funcionamento do protótipo de *skill*, a partir disto temos um questionamento a respeito do acompanhamento psicológico, a fim de armazenar esta informação para que ao longo do uso da *skill* possa se indicar a retomada ou iniciação deste processo de acompanhamento profissional. Dando sequência, os assuntos a serem abordados são divididos em 3 classes: conteúdos de psicoeducação, autocontrole e resolução de problemas. Dentro de cada classe haverá diversas

abordagens com o intuito de instruir os usuários a tentarem algumas reflexões para modificação comportamental e até mesmo algumas ações corriqueiras para obter melhores resultados em seus dias. Em meio a alguns assuntos haverá pequenas interações personalizadas para tornar o diálogo mais orgânico e menos engessado.

## 5.1 DIÁLOGOS / TEMAS

Os diálogos desenvolvidos buscaram abordar diferentes assuntos dentro de cada um dos temas escolhidos da Terapia Cognitivo Comportamental. Inicialmente foram buscados conteúdos mais genéricos e de senso comum à pessoas que possuam depressão, a fim de atender uma gama maior de pessoas e não adentrar em áreas muito profundas da psicologia. Em uma nova conversa com a psicóloga Cláudia, foram debatidos quais assuntos dentro dos temas escolhidos seriam interessantes entrar no desenvolvimento do trabalho e que tivessem uma boa relevância para a área. Com esta finalidade, houve a recomendação da leitura de dois artigos para uma melhor apropriação de conteúdos e formas de se abordar uma conversa com uma pessoa com depressão. São eles:

1. Terapia cognitivo-comportamental da depressão.
2. Manual de Terapias Cognitivo Comportamentais.

Powell et. al (2008) e Dobson (2006) afirmam que as estratégias terapêuticas da TCC da depressão envolvem trabalhar três pilares:

1. Pensamentos automáticos e episódios depressogênicos.
2. Estilo da pessoa se relacionar com os outros.
3. Mudança de comportamento a fim de obter melhor enfrentamento da situação problema.

Dobson (2006) ainda reforça a importância da psicoeducação como ferramenta no auxílio ao tratamento à depressão, uma vez que os pacientes podem aprender habilidades para modificar pensamentos e com isso ter maior controle sobre as emoções e condutas mais assertivas. Deve-se atuar sempre de forma colaborativa para que o paciente possua responsabilidades e metas, de forma a aumentar sua autoeficácia e confiança para as próximas situações de desafio.

Com base nas leituras foram definidos três assuntos para o tema Psicoeducação (Prioridades, Empatia e Hábitos), dois para Autocontrole (Emoções e Pensamentos Automáticos) e dois para Resolução de Problemas (Frustrações e

Procrastinação). Os assuntos de psicoeducação se propõem a auxiliar a pessoa a lidar melhor com as situações, ter um estilo de vida que traga maior bem estar e melhor relacionamento interpessoal. Já os conteúdos de auto controle possuem um cunho mais investigativo nos pensamentos e emoções e o próprio entendimento sobre a relação que eles possuem. Em resolução de problemas há um foco em colocar o paciente em ação e solidificar esse hábito, ao mesmo tempo que desconstrói crenças da própria pessoa sobre não se julgar capaz ou fugir de problemas. Sendo assim os diálogos instrutivos criados ficaram da seguinte forma:

### **Figura 12 - Conteúdo de pensamentos automáticos**

#### **Pensamentos automáticos**

Os pensamentos automáticos são aqueles pensamentos que brotam na nossa mente sem que tenhamos controle ou reflexão sobre eles. Como um pastel de queijo... hummmm... quer dizer, não esse tipo de pensamento automático heheh mas pensamentos que são disparados com algumas situações e que sempre se repetem continuamente, todas as pessoas tem pensamentos automáticos, mas existem alguns que são negativos e podem te atrapalhar, por isso vamos tentar identificar esses pensamentos e tentar muda-los para que não gerem mais desconforto. A primeira dica que vou te dar é que você não tente reprimir ou fugir destes pensamentos, como eu disse, todo mundo tem pensamentos automáticos, isso é comum, então o correto nessa situação seria aceitar o pensamento, ou melhor, acolher ele para assim poder identificar os seus motivos e modificar a forma de pensar. Eu tenho um truque que me ajudou muito nesse aspecto e já ensinei para várias pessoas que utilizaram a skill, você quer ouvir ?

Não -> aaah tem certeza ? Não -> Tudo bem, nos vemos amanhã

Sim -> Bom, a ideia é fazer como se fosse um diário de bordo, onde você vai anotar o pensamento automático que você teve, o sentimento que isso te gerou e depois você vai pensar e escrever pensamentos alternativos a essa mesma situação que te gerou o pensamento automático. Você pode fazer 3 colunas se quiser, uma do pensamento automático, depois o sentimento gerado, e por fim, alternativas diferentes de pensamento a respeito dessa situação. Dessa forma você vai começar a flexibilizar a sua forma de pensar, para parar de ir sempre para um mesmo caminho. Não se esqueça de ir anotando sempre que esses pensamentos acontecerem. Vou estar torcendo por você, amanhã voltamos a nos falar.

Fonte: Elaborado pelo Autor (2020)

Na figura 12 temos o conteúdo de pensamentos automáticos, um dos mais relevantes nas leituras feitas sobre o assunto. Podemos observar que há uma contextualização com caráter instrutivo sobre o assunto, junto de um questionamento

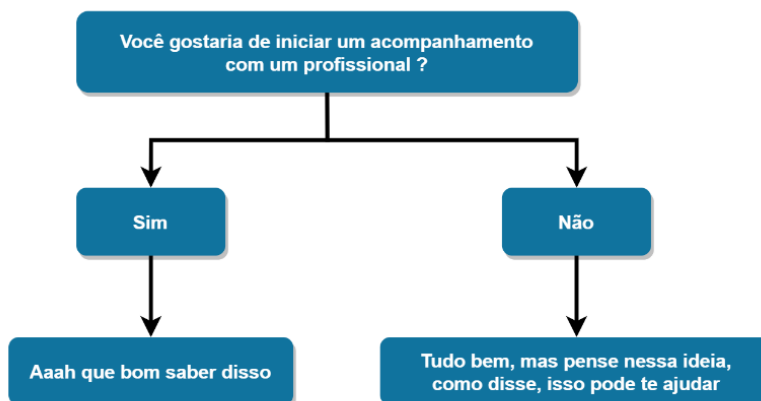


para dar continuidade caso seja do interesse do usuário. Esta interação no meio da fala tem o objetivo de quebra da escuta, para que não se torne muito monótono e o usuário sinta-se mais envolvido no uso da *skill*. Dando continuidade, é passada uma versão simplificada do Registro de Pensamentos Disfuncionais (RPD) citado por Dobson (2006) em seus estudos como uma das técnicas amplamente utilizadas para identificação e modificação de pensamentos automáticos. Técnica essa que permite a melhora da identificação destes pensamentos e a flexibilização cognitiva ao estimular o paciente a pensar em formas alternativas de pensamento.

Os diálogos iniciais tiveram uma validação pela psicóloga Claudia, que apontou que a forma como foram construídos ficou adequada a proposta, principalmente com o contraponto da personalidade da Alexa. Também foi ressaltado que foram escolhidos bons assuntos a serem abordados. Este contato teve o único intuito de alinhar se a forma de criação dos diálogos estava de acordo com o atendimento do psicólogo a pessoas com depressão. Foi salientado ainda pela psicóloga Claudia, a importância de selecionarmos bem as palavras a serem usadas, para não soar de forma hostil ou que induza o usuário a ter pensamentos negativos. Sempre criar diálogos leves e positivos, sem serem muito extravagantes.

Seguindo nesta linha foram desenvolvidos os demais assuntos a serem explorados pelo protótipo. Os conteúdos e fluxo completo do funcionamento do protótipo encontram-se no link público do Google Drive para melhor visualização <https://drive.google.com/file/d/1D5OnkAZRgn7vdD4cPhdsslo1xCo11HBD/view?usp=sharing> bem como no apêndice A deste trabalho. Como mencionado anteriormente, ao longo do uso da *skill* é sugerido ao usuário que retorne ou inicie um acompanhamento com um profissional da área, as perguntas e respostas são baseadas na sua resposta inicial sobre o seu histórico de acompanhamento. Na figura 13 temos um dos casos de sugestão.

Figura 13 - Sugestão de acompanhamento



Fonte: Elaborado pelo Autor (2020)

As interações vão acontecendo até que todos os assuntos de TCC tenham chegado ao seu fim, quando isso ocorrer há uma mensagem de fim da *skill*, onde o usuário é orientado a dar continuidade com um profissional.

## 5.2 APLICAÇÃO DA VUI

De acordo com os estudos realizados a respeito de VUI, foi pensado como explorar os pontos mais relevantes desta interação por voz, de uma forma com que os diálogos pudessem abordar os temas ao ponto de cumprirem seu papel no caráter instrutivo de forma clara e ao mesmo tempo focando em reter os usuários e não tornar a experiência com a *skill* negativa.

Deste modo, foram aplicadas principalmente as características de flexibilização de entradas, personalidade, elementos do princípio cooperativo, continuidade e fluidez. No que diz respeito a flexibilização de entradas está demonstrado de forma mais ampla no subcapítulo 5.4 de desenvolvimento do protótipo, que contém a exibição das entradas de respostas que o usuário pode fazer e sua elaboração.

Os diálogos tiveram grande foco na personalidade da Alexa, para que o usuário sentisse que está falando com o personagem original da assistente virtual e reforçasse o sentimento de compreensão na interação. Para isso foi mantida uma informalidade nas falas com o objetivo de aproximar essa interação de uma conversa comum entre pessoas, pois nos comunicamos de maneira mais coloquial quando verbalmente. Ainda na linha da personalidade da assistente, foi adicionado um tom como se ela possuísse sentimentos, para humanizar ao máximo o diálogo e confortar o usuário,

tendo em vista que deve-se ter muito cuidado na seleção das palavras a serem utilizadas com pessoas em depressão. O sentimento de compreensão é muito importante nesse aspecto.

Todos os princípios cooperativos foram utilizados na construção das interações entre a assistente e o usuário, sendo o elemento de quantidade o elo mais fraco nesta construção, pois o caráter instrutivo da *skill* necessita que haja uma explicação maior para que o usuário possa compreender os conceitos fundamentais das técnicas da terapia cognitivo comportamental. A personificação da Alexa também foi outro ponto que contribuiu para que os diálogos se tornassem um pouco mais extensos, contudo, ainda foi aplicada esta característica aos diálogos, tentando mantê-los sucintos dentro do possível, de modo a não impactar de forma expressiva em outros atributos de VUI da *skill*.

Outras medidas aplicadas foram a continuidade e fluidez junto da variedade das respostas dadas pela Alexa. Para tanto, procuramos elaborar respostas que sempre entregassem um pouco a mais e não somente uma confirmação ou feedback pouco elaborado. Todas as respostas procuraram utilizar destes princípios. Um bom exemplo da variedade aplicada no protótipo encontra-se na saudação de retorno à *skill*, onde há uma lista de possíveis falas da Alexa e sempre que o usuário retorna à habilidade é sorteada uma destas opções contidas em um *array*.

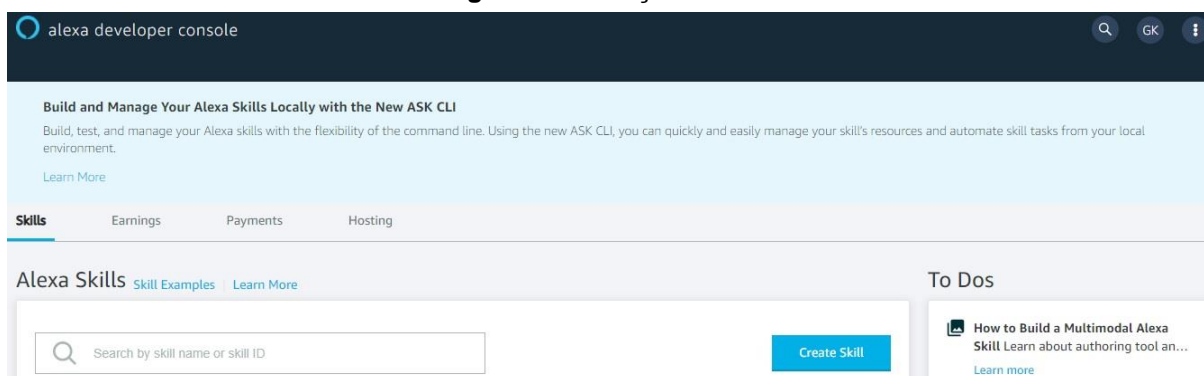
### 5.3 CURSO AMAZON

Com o objetivo de apropriar-se melhor da criação e desenvolvimento de *skills* bem como compreender o funcionamento da assistente virtual como um todo, foi realizado o curso disponibilizado pela Amazon para as pessoas que pretendem desenvolver *skills* para a Alexa, o “Como desenvolver uma *skill* de Alexa envolvente” que compreende 9 módulos básicos para criar uma habilidade, nele são abordados desde alguns conceitos de VUI, detalhamento do funcionamento da Alexa, até a publicação na loja de *skills*. Este treinamento forneceu a base para que pudesse ser desenvolvido o protótipo planejado, além disso a Amazon possui toda a documentação do Alexa Skills Kit em seu site para desenvolvedores, lá encontram-se algumas explicações adicionais sobre o assunto.

## 5.4 DESENVOLVIMENTO DO PROTÓTIPO

Para iniciar o desenvolvimento do protótipo de *skill* utilizando o Alexa Skills Kit, foi necessário primeiramente criar uma conta Amazon e acessar o console de desenvolvedor através do link <https://developer.amazon.com/alexa/console/ask>. Onde tanto são criadas novas *skills*, quanto listas as existentes na conta, conforme podemos observar na figura 14.

Figura 14 - Criação de skill



Fonte: Elaborado pelo Autor (2020)

Clicando no botão “Create Skill” os próximos passos foram definir um nome público para a *skill*, ele será usado para a busca na *store* posteriormente pelos usuários que desejarem fazer sua utilização. Esse nome não precisa ser relacionado a chamada da *skill* e nem afetará o seu uso, ele se dá exclusivamente para a busca na loja de habilidades e por isso desempenha um papel importante no ranking de resultados, junto de classificação, descrição e categoria. É importante que o nome não seja muito genérico para que a habilidade tenha um destaque das outras e remeta algo exclusivo ao usuário, por esse motivo o nome público definido foi “Inside”. Também foi definido o idioma padrão da *skill* como Português(BR) e a linguagem de programação a ser utilizada para manipular o *back-end* como Node.js conforme a figura 15.

Figura 15 - Definições criação de skill

alex developer console

## Create a new skill Cancel

Home

**Skill name**

Inside 6/50 characters

**Default language**

Portuguese (BR) v

More languages can be added to your skill after creation

### 1. Choose a model to add to your skill

There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.

**Custom** SELECTED

Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.

**Flash Briefing**

Give users control of their news feed. This pre-built model lets users control what updates they listen to.

"Alexa, quais são as notícias?"

**Smart Home**

Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.

"Alexa, acenda as luzes da cozinha"

**Video**

Let users find and consume content. This pre-built model supports content search and content suggestions.

"Alexa, reproduza Inter"

### 2. Choose a method to host your skill's backend resources

You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.

**Alexa-Hosted (Node.js)** SELECTED

Alexa will host skills in your account up to the AWS Free Tier limits and get you started with a Node.js template. You will gain access to an AWS Lambda endpoint, 5 GB of memory, and 1000 requests per second.

**Alexa-Hosted (Python)**

Alexa will host skills in your account up to the AWS Free Tier limits and get you started with a Python template. You will gain access to an AWS Lambda endpoint, 5 GB of memory, and 1000 requests per second.

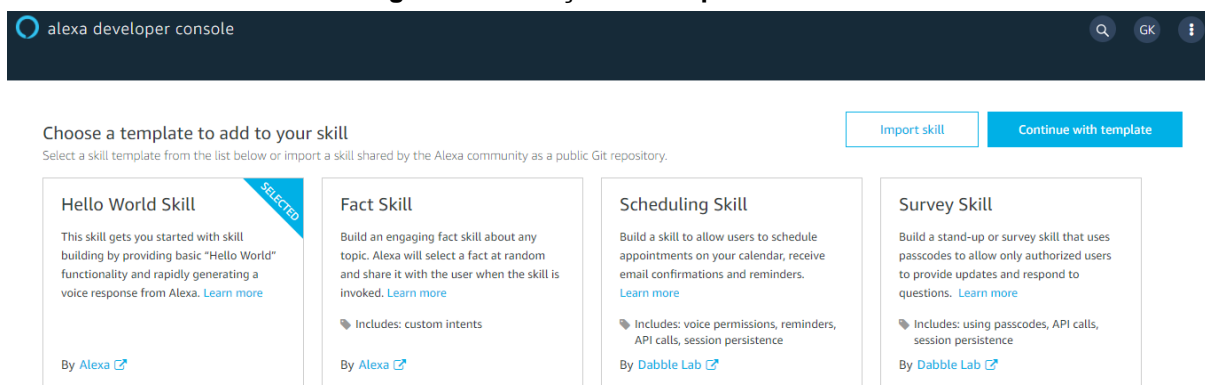
**Provision your own**

Provision your own endpoint and backend resources for your skill. This is recommended for skills that have significant data transfer requirements. You will not gain access to our code editor.

Fonte: Elaborado pelo Autor (2020)

Caso tenha sido escolhido alguma linguagem de *back-end* já utilizada pela Amazon, ao avançar para o próximo passo haverá uma seleção de 4 *templates* de skills básicas que estão disponíveis no Git da Amazon, escolhendo algum modelo já é possível finalizar a criação da *skill*. Conforme a figura 16, foi escolhida a *Hello World Skill* para dar início ao desenvolvimento do protótipo.

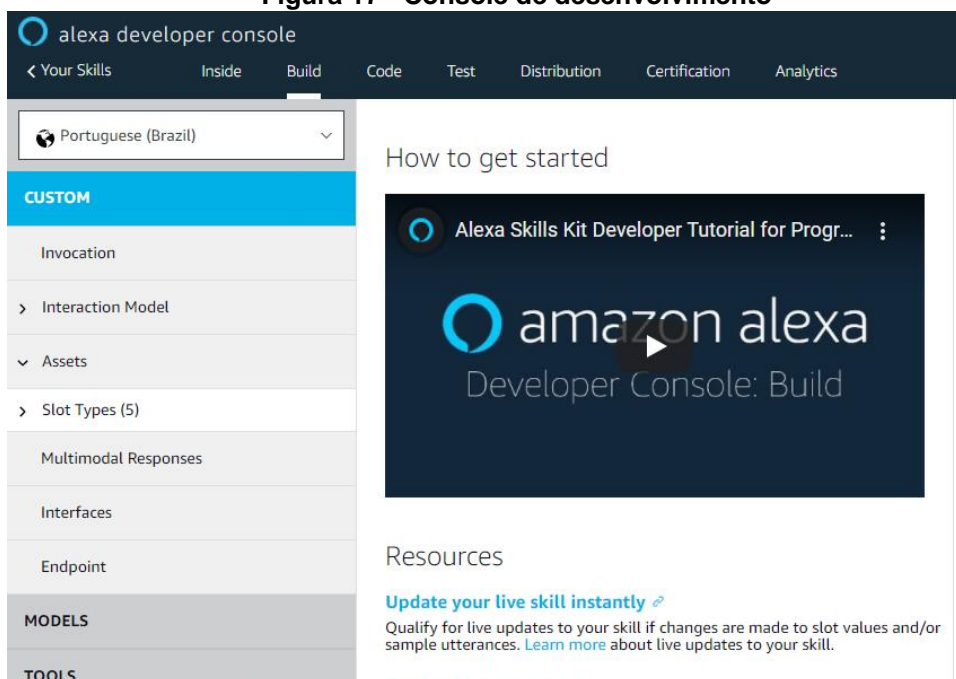
**Figura 16 - Seleção de template de skill**



Fonte: Elaborado pelo Autor (2020)

Com a criação da *skill* concluída, passa-se a ter acesso ao console de desenvolvimento, onde estão reunidas todas as ferramentas necessárias para que seja realizado o desenvolvimento. Na figura 17 é demonstrado este console de desenvolvimento do Alexa Skills Kit.

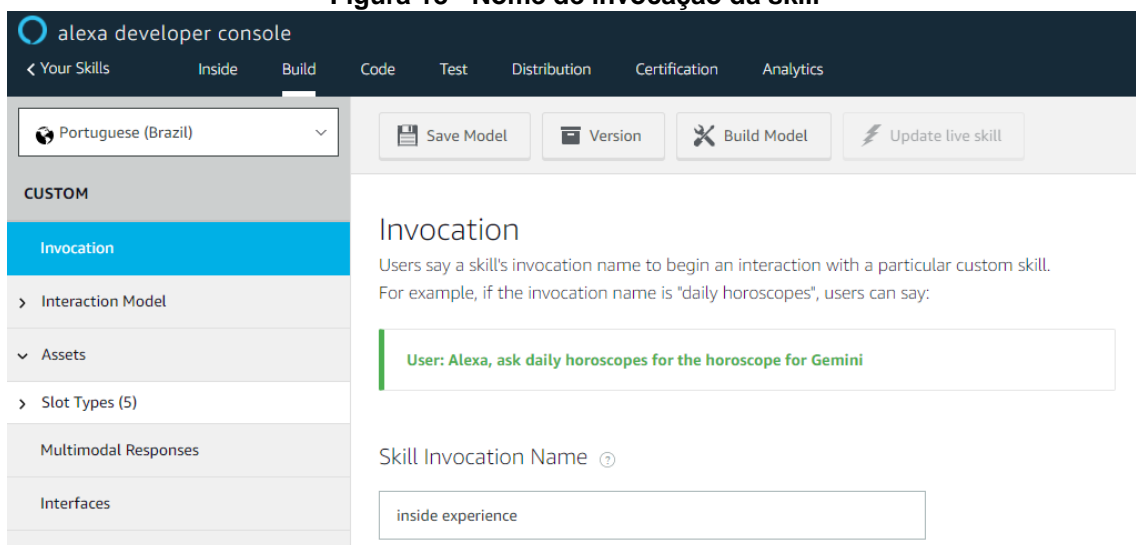
**Figura 17 - Console de desenvolvimento**



Fonte: Elaborado pelo Autor (2020)

Em seguida foi atribuído um nome de invocação para a *skill*, que foi definido como "Inside Experience", como podemos observar na figura 18. Este nome será utilizado pelo usuário para abrir a *skill*.

Figura 18 - Nome de invocação da skill



Fonte: Elaborado pelo Autor (2020)

Inicialmente existem 4 intenções fixas, que são requisito obrigatório para o funcionamento de qualquer habilidade, são elas:

AMAZON.CancelIntent

AMAZON.HelpIntent

AMAZON.StopIntent

AMAZON.NavigateHomeIntent

Pensando que a primeira interação da *skill* será perguntar ao usuário se ele tem ou já teve acompanhamento profissional no tratamento da depressão, a primeira intenção criada foi a *CapturaAcompanhamentoIntent*. Partindo do pressuposto que o usuário poderá responder de forma positiva ou negativa a esta pergunta. Foram criados 2 *slots* para comportar as diferentes expressões de afirmação ou negação, estes slots foram nomeados como *confirmacao* e *negacao*. Para o *slot confirmacao* foram atribuídos os valores:

“Sim”

“Estou”

“Já”

“Yes”

“Positivo”

E para o *slot negação* os valores:

“Não”

“Não estou”

“Ainda não”







“No”

“Negativo”

“Nunca”

Desta forma ainda são necessários diferentes enunciados para que a intenção esteja completa. Combinando os *slots* com as diferentes formas de resposta, obtivemos o modelo de intenção obtido demonstrado na figura 19.

**Figura 19 - Enunciados com slots**

{negacao} fazendo	
{negacao}	
{negacao} estou fazendo acompanhamento	
{confirmacao} fazendo	
{confirmacao} estou fazendo acompanhamento	
{confirmacao}	

Fonte: Elaborado pelo Autor (2020)

Para dar seguimento no desenvolvimento da função lambda, foi optado por utilizar a extensão Alexa Skills Toolkit For VS Code para utilização do framework Visual Studio Code na programação, devido a maior praticidade e ferramentas que o software proporciona para o desenvolvimento, principalmente quando comparado ao console online da Amazon. Os maiores benefícios desta escolha são: a possibilidade de edição offline, utilização de extensões do VS Code para auxiliar no desenvolvimento e melhor utilização do sistema de versionamento de código.

Para a realização do desenvolvimento do protótipo e testes de seu funcionamento, foi utilizado um exemplar da Alexa cedido pela universidade Feevale. Este exemplar corresponde ao modelo Echo, onde não é contemplada tela no dispositivo, apenas a interface de voz.



### 5.4.1 Função Lambda

A seguir na figura 20 temos o código inicial do `index.js` que é o arquivo principal do *back-end* da aplicação. Nele são encontrados todos os manipuladores de intenção, ou como são chamados, *handlers*. Os *handlers* normalmente são compostos em dois trechos principais, onde a primeira parte é feita uma verificação se o manipulador pode processar a solicitação feita. E a segunda parte é de fato o código para atender à esta demanda se for o caso.

Figura 20 - Handler de inicialização

```
// ASK SDK
const Alexa = require('ask-sdk-core');
// ASK SDK adapter para conectar ao Amazon S3
const persistenceAdapter = require('ask-sdk-s3-persistence-adapter');

//----- Inicialização / Saudação -----

const LaunchRequestHandler = {
  // Verifica se esse manipulador pode lidar com a solicitação
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchRequest';
  },
  // Código que lida com a solicitação
  handle(handlerInput) {
    // Cria sessão para atributos
    const sessionAttributes = handlerInput.attributesManager.getSessionAttributes();

    // Carrega atributos de sessão
    const confirmacao = sessionAttributes['confirmacao'];
    const negacao = sessionAttributes['negacao'];

    // Verifica se o usuário já respondeu a pergunta de acompanhamento e encaminha para a proxima etapa
    const acompRespondido = confirmacao || negacao;
    if (acompRespondido){
      await feedbackTratamento(handlerInput);
      return SorteioConteudosIntentHandler.handle(handlerInput);
    }

    // Mensagem introdutória da skill
    let speakOutput = handlerInput.t('Olá! Seja bem vindo a skill inside experience, é um prazer ter voc
+ ('Vou começar te explicando um pouco sobre esta skill.....')
+ ('Nosso objetivo é contribuir para a melhora da depressão, para que você passe por essa experiênci
const pergunta = handlerInput.t('Você está fazendo algum tratamento com psicólogo ou psiquiatra atua

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(pergunta)
      .getResponse();
  }
};
```

Fonte: Elaborado pelo Autor (2020)

O primeiro *handler* desenvolvido é o de inicialização da *skill*, sempre que houver uma requisição `LaunchRequest` este manipulador é invocado. Ele tem um funcionamento bem simples, carrega os atributos de sessão que foram salvos no

banco de dados e caso algum esteja preenchido, o que significa que já houve uma sessão anterior onde o usuário respondeu à pergunta inicial, há um redirecionamento para outro manipulador que irá dar sequência no fluxo da *skill*. Caso contrário, é dita a mensagem de boas-vindas, juntamente do questionamento referente ao acompanhamento profissional e seus respectivos feedbacks à cada resposta.

No final de todo *handler* é necessário retornar uma resposta para a requisição, para isto, são encapsulados os valores em funções de um construtor de resposta, para que o SDK execute o código e o retorne à requisição. Deste modo, passamos a mensagem de introdução nomeada como “speakOutput” para a função `.speak()`, que é responsável pela fala da Alexa, porém se usada de forma isolada, após a fala a habilidade é encerrada, por isso utilizamos também a função `.reprompt()`, que é responsável por ouvir uma resposta do usuário. Neste caso ainda há a possibilidade de utilizar-se desta função para falar algo ao usuário mais uma vez, por isso essa função recebe a pergunta de acompanhamento para que seja dita antes que o manipulador se encerre.

Cada handler permite apenas a utilização de um `.speak()` e um `.reprompt()`, ou seja, é possível falar algo ao usuário até duas vezes por manipulador e ouvir apenas uma vez. Podemos observar que também há uma função chamada `.getResponse()` que não recebe nenhum parâmetro, isso se dá porque ela é responsável por transformar todo o conjunto de falas em uma única resposta JSON para a *skill*.

Ao final do `index.js` há um construtor da *skill* onde é preciso adicionar os *handlers* criados. Se estes *handlers* forem relacionados a uma intenção é obrigatório que constem no construtor da *skill*, caso contrário o projeto será compilado com erro. Porém, também há a possibilidade de utilizar *handlers* independentes de intenções, estes não possuem a necessidade de declaração. Na figura 21 podemos observar que são adicionados os *handlers* das intenções obrigatórias, de inicialização da *skill* e o relativo a intenção criada anteriormente para capturar as respostas do usuário.

Figura 21 - Construtor de skill

```
exports.handler = Alexa.SkillBuilders.custom()  
  .withPersistenceAdapter(  
    new persistenceAdapter.S3PersistenceAdapter({ bucketName: process.env.S3_PERSISTENCE_BUCKET })  
  )  
  .addRequestHandlers(  
    LaunchRequestHandler,  
    CapturaAcompanhamentoIntentHandler,  
    HelpIntentHandler,  
    CancelAndStopIntentHandler,  
    SessionEndedRequestHandler,  
    IntentReflectorHandler)  
  .addErrorHandlers(  
    ErrorHandler)  
  .withApiClient(new Alexa.DefaultApiClient())  
  .lambda();
```

Fonte: Elaborado pelo Autor (2020)

Como o *handler* de inicialização termina com uma pergunta, cujas respostas a esta pergunta estão contidas nos enunciados da intenção CapturaAcompanhamentoIntent como visto anteriormente, assim que o usuário pronunciar algum destes enunciados, a intenção será disparada chamando seu manipulador. Este por sua vez, já possui associado o nome da intenção que ele deve atender em seu cabeçalho, comparado com o *handler* de inicialização que somente verificava o tipo de requisição feita. Após isso, são capturados os valores dos *slots* e adicionados em variáveis que serão salvas no banco de dados como atributos de sessão.

Para este armazenamento em banco de dados, a Amazon disponibiliza o Amazon S3, que é um serviço de *storage* projetado para facilitar o desenvolvimento web. Para utilizar o S3 basta declarar no início do arquivo principal do *back-end* que é necessária a utilização do *adapter* do serviço e posteriormente adicioná-lo no construtor da *skill*. A partir disso já é possível salvar e buscar valores em banco.

Figura 22 - Handler de respostas parte 1

```

//----- Captura Dados Acompanhamento -----
const CapturaAcompanhamentoIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'CapturaAcompanhamentoIntent';
  },
  async handle(handlerInput) {
    const { attributesManager, requestEnvelope } = handlerInput;

    // Captura os valores obtidos nos slots
    const confirmacao = Alexa.getSlotValue(requestEnvelope, 'confirmacao');
    const negacao = Alexa.getSlotValue(requestEnvelope, 'negacao');
    var respondeu;

    // Cria as atributos de sessão
    const sessionAttributes = attributesManager.getSessionAttributes();
    const {intent} = requestEnvelope.request;

    // Carrega se o usuário já respondeu a pergunta uma vez
    const sessionAttributesCarrega = handlerInput.attributesManager.getSessionAttributes();
    var respondeuCarregado = sessionAttributesCarrega['respondeu'];

    // Salva atributos de sessão
    sessionAttributes['confirmacao'] = confirmacao;
    sessionAttributes['negacao'] = negacao;

    // Variáveis para conferência de não redundância (atributos de sessão carregados para dentro da variável)
    const slotConfirmacao = sessionAttributes['confirmacao'];
    const slotNegacao = sessionAttributes['negacao'];
    respondeu = false;
  }
};

```

Fonte: Elaborado pelo Autor (2020)

Em seguida, o próximo *handler* criado foi o `capturaAcompanhamentoIntentHandler` conforme a figura 22. Onde são feitas verificações de qual *slot* foi preenchido e qual a resposta dada pelo usuário para o questionamento do acompanhamento profissional, para que assim possa dar o feedback correto a ele. Como a intenção é acionada toda vez que o usuário responder algo que esteja nos enunciados/textos de exemplo da intenção, são utilizadas *flags* salvas em banco, junto de encaminhamentos para os manipuladores correspondentes à situação, a fim de evitar que a *skill* repita frases já respondidas. Desta mesma forma este manipulador é utilizado para receber todas as respostas que o usuário dará a Alexa. Pois todas elas partem do princípio de serem positivas ou negativas, então a *skill* se detêm ao uso desta única intenção para efetuar essa captura de respostas.

Figura 23 - Hanlder de respostas parte 2

```

// Verificações de caso
if(slotNegacao){
  // Respondeu que não está fazendo tratamento
  if(respondeuCarregado !== true){...
}
}
else{
  // Questões de Pensamentos Automáticos
  if(respPensaAutCarregado === true){...
}
else{
  // Questão de Frustrações
  if(respFrustCarregado === true && respFrustOkCarregado !== true){...
}
else{
  // Respondeu que não gostaria de VOLTAR ao acompanhamento profissional
  if(RespTeveAcompCarregado === true){...
}
  // Respondeu que não gostaria de COMEÇAR acompanhamento profissional
  else if(RespNuncaTeveAcompCarregado === true){...
}
  // Respondeu que não está GOSTANDO acompanhamento profissional
  else if(RespTemAcompCarregado === true){...
}
}
else{
  // Se respondeu não a pergunta de tratamento
  sessionAttributes['temAcompanhamento'] = false;
  sessionAttributes['jaTeveAcompanhamento'] = false;
  await feedbackTratamento(handlerInput);
  return SorteioConteudosIntentHandler.handle(handlerInput);
}
}
}

```

Fonte: Elaborado pelo Autor (2020)

Podemos observar na figura 23 que a partir do *slot* preenchido, existe a identificação de qual pergunta o usuário respondeu para que assim, possa ser feito o processamento adequado à situação.

Caso se trate das questões iniciais de acompanhamento, é utilizada uma *function* para dar o feedback ao usuário e posteriormente o encaminha para o *handler* de sorteio de conteúdos. Que por sua vez possui uma lista dos conteúdos que o protótipo abrange e efetua o sorteio de forma aleatória. Toda via, é verificado no banco de dados se é a primeira vez que o tema é sorteado, com a finalidade de evitar a repetição indesejada. O trecho de código citado é exibido na figura 24 abaixo.

Figura 24 - Handler de sorteio de conteúdos

```
// Sugere que o usuário considere iniciar ou retomar o tratamento com um profissional
if(assuntosJaSorteados.length === 3 && respAcompOk !== true || assuntosJaSorteados.length === 6
  return sugestaoAcompanhamento.handle(handlerInput);
}else{
  sessionAttributes['respAcompOk'] = false;

  const arrayAssuntos = [
    /*1*/ 'Prioridades',
    /*2*/ 'Empatia',
    /*3*/ 'Hábitos',
    /*4*/ 'Emoções',
    /*5*/ 'Pensamentos Automáticos',
    /*6*/ 'Frustrações',
    /*7*/ 'Procrastinação'
  ];
  var assuntoIndex = Math.floor(Math.random() * arrayAssuntos.length);
  var assuntoSorteado = arrayAssuntos[assuntoIndex];

  // Se ainda houver assuntos remanescentes
  if(assuntosJaSorteados.length < arrayAssuntos.length){
    // Enquanto for sorteado um assunto que já foi sorteado antes, continua sorteando
    while(assuntosJaSorteados.find(element => element === assuntoSorteado)){
      assuntoIndex = Math.floor(Math.random() * arrayAssuntos.length);
      assuntoSorteado = arrayAssuntos[assuntoIndex];
    }
    // Adiciona ao array o assunto sorteado
    assuntosJaSorteados.push(assuntoSorteado);

    // Salva atributos de sessão
    sessionAttributes['assuntosJaSorteados'] = assuntosJaSorteados;

    if(assuntoSorteado === 'Prioridades'){
      return PrioridadesIntentHandler.handle(handlerInput);
    }
    else if(assuntoSorteado === 'Empatia'){
      return EmpatiaIntentHandler.handle(handlerInput);
    }
  }
}
```

Fonte: Elaborado pelo Autor (2020)

Em seu início também há a verificação de quantos conteúdos o usuário já passou até o momento, para que ao atingir a quantidade de 3 e 6 temas abordados, seja feita a sugestão de acompanhamento profissional. Por se tratar de um protótipo, o escopo de temas é em menor escala, desta maneira, como existem 7 temas abordados durante o protótipo, foram definidas posições fixas onde ocorrerá a sugestão de acompanhamento.

Após o sorteio, há a identificação de qual *handler* está associado ao conteúdo sorteado. Desta forma a aplicação explicará o seu conteúdo específico, juntamente de suas respectivas interações se for o caso, subsequentemente é dita uma mensagem de despedida e a *skill* é encerrada.

Caso todos os assuntos já tenham sido sorteados é dita a mensagem final da *skill* ao usuário, onde o orienta que continue o acompanhamento com um profissional, tendo em vista que a *skill* chegou ao seu fim.

**Figura 25 - Hanlder de conteúdo específico**

```
// ----- Pensamentos Automáticos -----

const PensamentosAutomaticosIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
  },
  async handle(handlerInput) {

    // Carrega assuntos já abordados
    const sessionAttributesCarrega = handlerInput.attributesManager.getSessionAttributes();

    let speakOutput = handlerInput.t('');
    speakOutput += '<break time="1000ms"/> Hoje vamos falar sobre Pensamentos Automáticos <break time="1000ms"/>';
    speakOutput += '<prosody rate="90%">Os pensamentos automáticos <break time="100ms"/> são aqueles pensamentos que |
+ 'sobre eles. <break time="300ms"/> Como um pastel de queijo...<break time="300ms"/> hummm... <break time="300m
+ 'com algumas situações <break time="300ms"/> e que sempre se repetem continuamente. Todas as pessoas tem pensam
+ 'e podem te atrapalhar. <break time="100ms"/> Por isso vamos tentar identificar esses pensamentos e tentar mudã
+ 'dica que vou te dar <break time="100ms"/> é que você não tente reprimir ou fugir destes pensamentos, <break ti
+ 'e isso é comum <break time="200ms"/> então o correto nessa situação seria aceitar o pensamento,<break time="20
+ 'e modificar a forma de pensar.<break time="200ms"/> Eu tenho um truque que me ajudou muito nesse aspecto <brea
+ 'você quer ouvir ? </prosody>'

    sessionAttributesCarrega['respPensaAut'] = true;

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt()
      .getResponse();
  },
};
```

Fonte: Elaborado pelo Autor (2020)

Na figura 25 é apresentado um *handler* de conteúdo específico, exemplificando como se dá o funcionamento da *skill* após um conteúdo ser sorteado e encaminhado para seu manipulador correspondente.

No exemplo utilizado, primeiramente é explicada a parte instrutiva do conteúdo e ao seu fim, há um questionamento ao usuário para quebrar um pouco o excesso de escuta por se tratar de um assunto mais extenso. Por este motivo, há um `.reprompt()` associado ao manipulador, para que assim a Alexa possa ouvir a resposta e consequentemente entrar no `CapturaAcompanhamentoIntentHandler` e seguir o fluxo de diálogo que o usuário desejar. Em alguns casos devido a necessidade de um jogo de perguntas e respostas, obrigou-se a utilização de mais de um *handler* para englobar todo o diálogo desejado, tendo em vista os limites de utilização de fala e escuta que a aplicação dispõe.

Em dado momento, o número de conteúdos já abordados atinge os valores definidos para que haja a sugestão de acompanhamento, então há um

redirecionamento para o manipulador responsável. Na figura 26 é apresentado o código do manipulador referido.

**Figura 26 - Handler de sugestão de acompanhamento**

```
// ----- Sugestão de Acompanhamento -----
const sugestaoAcompanhamento = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
  },
  async handle(handlerInput) {

    // Carrega respostas anteriores
    const sessionAttributesCarrega = handlerInput.attributesManager.getSessionAttributes();
    var jaTeveAcompanhamentoCarregado = sessionAttributesCarrega['jaTeveAcompanhamento'];
    var temAcompanhamentoCarregado = sessionAttributesCarrega['temAcompanhamento'];

    let speakOutput = handlerInput.t('');

    // Se já fez acompanhamento/tratamento alguma vez
    if(jaTeveAcompanhamentoCarregado === true){
      speakOutput += 'Você gostaria de retornar ao acompanhamento profissional?';
      // Cria as atributos de sessão
      const sessionAttributes = handlerInput.attributesManager.getSessionAttributes();

      // Salva em banco que a resposta necessária será a respeito desse handler
      sessionAttributes['RespTeveAcomp'] = true;
      sessionAttributes['respAcompOk'] = false;
    }
    // Nunca fez acompanhamento/tratamento
    else if(temAcompanhamentoCarregado === false && jaTeveAcompanhamentoCarregado === false){
```

Fonte: Elaborado pelo Autor (2020)

O questionamento é baseado nas respostas dadas inicialmente pelo usuário, assim como o feedback desta pergunta posteriormente. Para a obtenção da resposta, mais uma vez é utilizada a intenção `CapturaAcompanhamentoIntent`, que retorna ao usuário uma resposta adequada a sua escolha e subsequentemente o encaminha para o sorteio de conteúdos novamente para seguir o fluxo normal.

Em situações de retorno a *skill*, o usuário é recebido com uma mensagem de boas-vindas personalizada, tendo seu funcionamento do mesmo modo que acontece com o sorteio de conteúdos. Há uma lista de mensagens receptivas onde é sorteada uma das opções de forma aleatória e dita ao usuário, neste caso não há controle das mensagens já abordadas, permitindo assim a repetição.



### 5.4.2 SSML

Ao se executar os diálogos foi observado que algumas pronúncias eram ditas erroneamente pela Alexa e que para ter uma melhor interpretação dos textos seria interessante fazer algumas pausas estratégicas e até mesmo falar de forma mais lenta para uma melhor absorção dos conteúdos. Para isto foi utilizada a linguagem SSML (*Speech Synthesis Markup Language*) da Amazon, que serve para gerenciar as saídas de voz e pronúncias da Alexa. Em todo o protótipo foram utilizadas apenas 4 *tags* das mais de 16 disponíveis, cada *tag* ainda possui parâmetros próprios que podem variar a escala aplicada ou alterar alguns detalhes sobre a funcionalidade em questão. As *tags* empregadas são exibidas na figura 27.

Figura 27 - Tags SSML aplicadas

```
<break time="400ms"/>  
<prosody rate="90%">Olá</prosody>  
<emphasis level="moderate">escrever</emphasis>  
<phoneme alphabet="ipa" ph="heheh">heheh</phoneme>
```

Fonte: Elaborado pelo Autor (2020)

Onde a *tag break* é utilizada para fazer pausas de determinado período durante a fala. *Prosody* permite alterar a velocidade da pronúncia da Alexa através de modelos pré-definidos ou de customização manual dessa velocidade. Também temos a *tag emphasis* para dar ênfase a uma palavra ou frase, podendo variar sua intensidade de acordo com o desejado. E por fim a *tag phoneme* para a pronúncia correta de fonemas, para tal é utilizado o alfabeto IPA, que é o alfabeto mundial de fonemas.

### 5.4.3 Dificuldades

Devido a Alexa ser uma assistente virtual, ou seja, criada com foco em executar tarefas, ela possui a característica de funcionar de forma assíncrona, por exemplo, a qualquer momento o usuário pode dizer “Alexa, acenda as luzes” e por isto as intenções podem ser acionadas independente de contexto.

Como o case do trabalho é ter um diálogo, pressupõe-se que a conversa deva seguir uma cronologia e não entrar em certos manipuladores ou intenções fora do momento correto. Então para contornar essa característica assíncrona foi necessário

utilizar *flags* salvas em banco para a sinalização de estado. Separando assim as abordagens já executadas e as que não foram requisitadas no momento, isto acabou garantindo a sincronicidade dos diálogos na aplicação. Tanto quanto redirecionamentos para os manipuladores desejados.

Ainda a respeito da assincronicidade, outro aspecto complicador foi a execução dos métodos de forma não sequencial, todo o código do *handler* era executado antes de surgir a fala ou a escuta por parte da Alexa, dificultando assim o desenvolvimento da aplicação. Pois fazia-se necessário a utilização de mais manipuladores para concluir o objetivo proposto.

Outra característica limitada da Alexa foi o fato de cada *handler* só poder executar um `.speak()` e um `.reprompt()`, o que obrigou à utilização um serviço de diretiva para que a Alexa pudesse ter mais falas em um único *handler*, para isso foi criada uma *function* e utilizado o *await* nesta função, desta forma a diretiva consegue fazer a Alexa falar mais de uma vez em um mesmo manipulador e executar um próximo passo apenas depois que esta função for executada. Porém a limitação de ouvir mais respostas do usuário em um mesmo *handler* permaneceu, tendo em vista que o serviço de diretiva não dá suporte a escuta, somente a fala.

## 5.5 VALIDAÇÃO DO PROTÓTIPO

Para validar a proposta apresentada neste trabalho, foram convidadas três professoras do curso de psicologia da universidade Feevale e uma psicóloga clínica voluntária para participar da validação via questionário online.

Devido a *skill* não estar publicada na *Alexa Skill Store*, sua execução necessita de um dispositivo Alexa com a conta do desenvolvedor vinculada. Para a demonstração do funcionamento do protótipo aos profissionais e levando em consideração o momento atual da pandemia do covid-19, optou-se por gravar um vídeo demonstrando a execução completa do protótipo desenvolvido. Abordando todas as rotas de seu funcionamento, porém com um escopo de conteúdos reduzido de 7 para 3 e com ajuste na sugestão de acompanhamento para após 1 tema abordado já efetuar o questionamento. Essa redução dos temas abordados em vídeo teve o objetivo de não tomar muito tempo dos profissionais entrevistados, por resultar em um vídeo muito extenso se abordado todo o escopo textual do protótipo. O vídeo referido encontra-se disponível na bibliografia de Knewitz (2020).

O questionário de validação constitui-se de 7 questões objetivas a respeito do protótipo como um todo (interações, utilização e contribuição) e 5 questões objetivas a respeito dos textos/conteúdos desenvolvidos. Foi utilizada a escala Likert como padrão para as respostas, onde 1 representa “discordo totalmente” e 5 “concordo totalmente”. O formulário ainda conta com uma questão dissertativa ao fim, para receber sugestões e opiniões com o intuito de identificar melhorias e falhas no protótipo. Na figura 28 temos as questões desenvolvidas para o formulário.

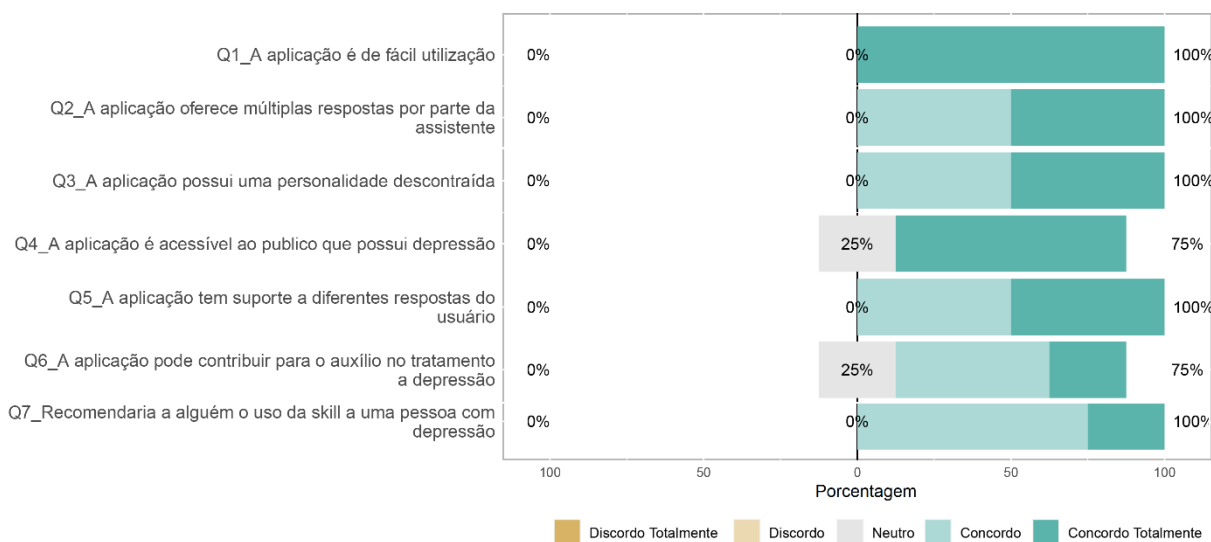
**Figura 28 - Questões do formulário de validação**

Critério de Análise	Id	Questão
Protótipo	Q1	A aplicação é de fácil utilização
	Q2	A aplicação oferece múltiplas respostas por parte da assistente
	Q3	A aplicação possui uma personalidade descontraída
	Q4	A aplicação é acessível ao público que possui depressão
	Q5	A aplicação tem suporte a diferentes respostas do usuário
	Q6	A aplicação pode contribuir para o auxílio no tratamento a depressão
	Q7	Recomendaria a alguém o uso da skill a uma pessoa com depressão
Conteúdos	Q8	Os textos condizem com temas abordados na psicologia
	Q9	Os textos produzem uma reflexão pessoal nos usuários
	Q10	Os textos são adequados para pessoas com depressão
	Q11	Os textos podem ajudar pessoas com depressão
	Q12	Os textos estão dentro dos temas abordados sobre depressão por profissionais
Sugestões	Q13	Sugestões de melhoria e comentários

Fonte: Elaborado pelo Autor (2020)

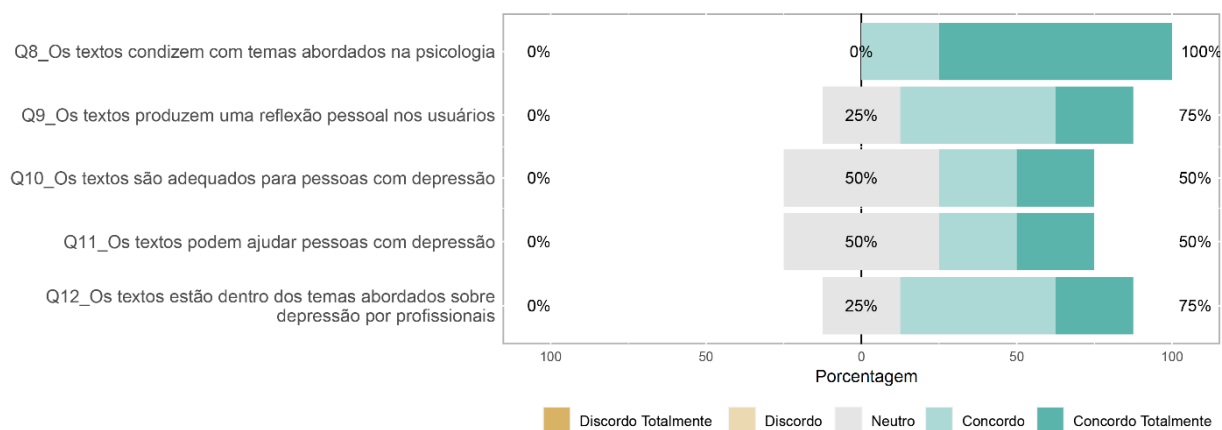
A composição do questionário aplicado pode ser consultada no Apendice B junto com a coleta dos dados feita ao final de seu preenchimento pelas partes envolvidas no processo de validação.

Para melhor interpretação dos dados obtidos, foi utilizado o software RStudio junto do pacote de gráficos likert para efetuar a plotagem dos dados e gerar uma visualização mais clara das informações. Na figura 29 são exibidos os resultados referentes as questões relacionadas ao protótipo/aplicação.

**Figura 29 - Resultados da validação da aplicação**

Fonte: Elaborado pelo Autor (2020)

Podemos observar de acordo com as respostas fornecidas, que as questões voltadas à aplicação em si, tiveram quase em sua totalidade respostas positivas, demonstrando assim a uma boa aceitação por parte das psicólogas. Observa-se também que as questões relacionadas a contribuição do protótipo no tratamento de depressão (Q6) e a acessibilidade da aplicação a este público (Q4) foram as perguntas que trouxeram uma maior dúvida às psicólogas envolvidas na validação, porém mesmo assim não houve respostas negativas quanto a aplicação. Ressalva-se ainda que a pergunta relacionada a indicação do protótipo a pessoas com depressão (Q7) obteve 100% de concordância, o que reforça a relevância do trabalho desenvolvido frente a doença.

**Figura 30 - Resultados da validação dos conteúdos**

Fonte: Elaborado pelo Autor (2020)

Conforme exposto na figura 30, os resultados obtidos pelas questões referentes aos conteúdos de psicologia desenvolvidos seguiram com um resultado semelhante ao apresentado pelas questões da aplicação. Porém, nesta seção do formulário já houve um maior posicionamento de forma neutra às questões.

As perguntas relacionadas a contribuição dos textos para a melhora da depressão (Q11) e a adequação destes conteúdos em um quadro de depressão (Q10) concentraram metade dos votos na posição “Neutro”. Mais uma vez, por não se apresentarem respostas negativas a respeito, podemos interpretar como um resultado positivo. Observando as respostas fornecidas em Q8, Q9 e Q12 é possível constatar que em um âmbito geral, as psicólogas entendem que os assuntos abordados possuem uma relevância na psicologia e que em sua maioria são temas abordados por profissionais que tratam a depressão, conforme evidenciado em Q12.

Quanto a questão de sugestões e comentários houve apenas duas respostas por parte das professoras da universidade Feevale. Os comentários estão descritos a seguir:

- Primeiro quero parabenizar o pesquisador pelo tema! De grande valia e penso que teremos uma interlocução cada vez maior com a IA e ela pode ser um excelente aliado no esclarecimento de questões e doenças. Uso a Alexa diariamente para várias questões do dia a dia. Os temas são todos ótimos e importantes, mas senti falta de uma conexão com a questão da depressão. A questão da depressão é difícil mesmo, pois se trata de um tema delicado e que envolve riscos. Desejo sucesso! abs;
- De maneira geral a aplicação é muito boa. Somente o tema empatia tratado, apesar de ser importante, não sei se é adequado a quem está em possível sofrimento, muitas vezes pela própria falta dela por parte de outros.

A partir dos comentários feitos é possível perceber que mesmo com estudos a respeito do tema, ainda há divergências dos conteúdos com o que alguns profissionais aplicam. Tendo em vista que a depressão é um tema delicado e amplo, o ideal seria a participação ativa de um psicólogo para desenvolver os textos e aproximar assim o nível de concordância da área. De forma geral a validação do protótipo e conteúdos se mostrou positiva, atendendo assim os objetivos propostos inicialmente no estudo.

## 6 CONCLUSÃO

Com base nos estudos realizados identificou-se a oportunidade de utilização da Alexa para contribuir no tratamento de depressão, bem como a atenção que a depressão deve receber no Brasil devido aos seus números em relação a doença.

Inicialmente, foi realizado um levantamento bibliográfico para identificação de técnicas psicológicas que pudessem ser aplicadas no protótipo. Neste processo, também foi realizado o estudo da parte de interface de voz e da própria assistente virtual que foi utilizada. Além disso, houve reuniões com a psicóloga Claudia Maria Teixeira Goulart, que atua como professora de psicologia na Universidade FEEVALE e nestas reuniões pôde-se verificar materiais necessários para o estudo, sugestões de funcionalidades para o protótipo e adequação dos temas com a proposta do trabalho. Pôde-se perceber também a relevância de técnicas como a psicoeducação e programas de auto ajuda frente a melhora da doença. Algo que contribui muito para o presente trabalho, por serem técnicas que utilizam mais a ação de mudança do próprio indivíduo, facilitando assim o trabalho do profissional com o paciente.

O protótipo desenvolvido visou incentivar o início ou retomada do tratamento de pessoas que sofrem de depressão através de diálogos instrutivos e técnicas comportamentais emergentes do estudo realizado. Concomitantemente de uma aproximação entre o paciente e o profissional de psicologia através da sugestão de acompanhamento e maior contato com o assunto. Em paralelo buscamos aplicar todas as orientações possíveis de VUI nos conteúdos e interações da aplicação, de forma a obter como resultado uma aplicação de voz bem estruturada e com uma boa aceitação dos usuários. Para avaliar a qualidade do protótipo desenvolvido foi feita uma validação através de questionário online com alguns profissionais da área. Nesta validação evidenciou-se um resultado positivo a respeito do trabalho desenvolvido, apenas com ressalvas quanto aos conteúdos que poderiam ter sido um pouco mais alinhados. Mas de forma geral não houve nenhum resultado negativo sobre o trabalho.

Durante o desenvolvimento apresentaram-se algumas dificuldades no âmbito de criação do protótipo. Os principais fatores para isto foram a assincronicidade que a Alexa possui tanto em relação a chamadas de intenções quanto na execução dos métodos, porém encontrou-se uma forma de contornar essa característica da assistente. Outra limitação que surgiu no desenvolvimento da aplicação foi a

quantidade restrita de interações que a Alexa possui, o que ocasionou o aumento da quantidade de recursos utilizados para se desempenhar as tarefas.

Ainda, percebem-se oportunidades de continuação do protótipo desenvolvido, como uma maior abrangência e alinhamento dos conteúdos abordados, a fim de aumentar a assertividade dos temas de psicologia e ter um maior engajamento do usuário com a *skill*. Além disso, a utilização de diferentes entonações quando disponíveis para o Brasil seriam um bom complemento para a habilidade, bem como a adição de funcionalidades pertinentes a depressão que possam ser elaboradas. É importante frisar ainda, que de acordo com os estudos realizados o uso de assistentes virtuais tende a crescer nos próximos anos, reforçando a tendência de utilização de *skills* de modo geral.

## REFERÊNCIAS BIBLIOGRÁFICAS

AMAZON. **Training Course: Build an Engaging Alexa Skill**. [2019?] Disponível em <<https://developer.amazon.com/en-US/alexa/alexa-skills-kit/get-deeper/tutorials-code-samples/build-an-engaging-alexa-skill/>>. Acesso em 02 de abril de 2020.

ANDRETTA, Ilana; OLIVEIRA, Margareth da Silva. **Manual Prático de Terapia Cognitivo-Comportamental**. Casa do psicólogo. São Paulo, 2011. 237 p.

BARCELLOS, Mário Tregnago; BURIGO, Ligia Marroni; AGOSTINHO, Milena Rodrigues; RADOS, Dimitris Rucks Varvaki (UFRGS). **TeleCondutas - Depressão**. 2017. Disponível em <[https://www.ufrgs.br/telessauders/documentos/telecondutas/Telecondutas\\_Depressao\\_20170428.pdf](https://www.ufrgs.br/telessauders/documentos/telecondutas/Telecondutas_Depressao_20170428.pdf)>. Acesso em 28 de maio de 2020.

COATES, Dustin A. **Voice Applications for Alexa and Google Assistant**. Manning Publications, 2019. Shelter Island, New York. 264 p.

DASGUPTA, Ritwik. **Voice User Interface Design - Moving from GUI to Mixed Modal Interaction**. 2018. Hyderabad, Telangana, India. 114 p.

DESMAZIERES, Marion. (AMAZON). **Alexa Skills Kit agora disponível para o Brasil — Alexa Voice Service será lançado ainda este ano**. Abr de 2019. Disponível em: <<https://developer.amazon.com/blogs/alexa/post/e40296bb-130e-40c9-b2fb-4505094d6bca/alexa-skills-kit-expands-to-brazil-alexa-voice-service-coming-later-this-year>>. Acesso em 27 de março de 2020.

DOBSON, Keith S.. **Manual de Terapias Cognitivo Comportamentais**. ArtMed Editora, 2006. São Paulo, Brasil. 327 p.

GAO, Catherine (AMAZON). **Use New Alexa Emotions and Speaking Styles to Create a More Natural and Intuitive Voice Experience**. Nov de 2019. Disponível em: <<https://developer.amazon.com/en-US/blogs/alexa/alexa-skills-kit/2019/11/new-alexa-emotions-and-speaking-styles>>. Acesso em 18 de março de 2020.

IPROSPECT. **Latin America finds its Voice**. 2019. Disponível em: <<https://www.iprospect.com/en/ar/news-and-views/insights/latin-america-finds-its-voice/>>. Acesso em 22 de março de 2020.

KINSELLA, Bret (VOICEBOT). **Amazon Again Topped Q4 Global Smart Speaker Sales Followed by Google and Baidu According to Strategy Analytics. Smart Speaker Shipments Set New Record**. Fev de 2020. Disponível em: <<https://voicebot.ai/2020/02/17/amazon-again-topped-q4-global-smart-speaker-sales-followed-by-google-and-baidu-according-to-strategy-analytics-smart-speaker-shipments-set-new-record/>>. Acesso em 28 de março de 2020.



KNEWITZ, Gustavo Maier. **Protótipo de skill para auxiliar na depressão (concluído) - Validação**. 2020, (10 min). Disponível em: <<https://www.youtube.com/watch?v=1kA133Zhrq0>>. Acesso em 31 de outubro de 2020.

MARCUS, Marina; YASAMY, M. Taghi; VAN OMMEREN, Mark; CHISHOLM, Dan. **Depression, a global public health concern**. 2012. Disponível em <[https://www.who.int/mental\\_health/management/depression/who\\_paper\\_depression\\_wfmh\\_2012.pdf](https://www.who.int/mental_health/management/depression/who_paper_depression_wfmh_2012.pdf)>. Acesso em 23 de março de 2020.

NOGUEIRA, Carlos André; CRISOSTOMO, Kelly Nunes; SOUZA, Rafaela dos Santos; PRADO, Jessica de Macedo do. **A Importância Da Psicoeducação Na Terapia Cognitivo-comportamental**. Revista das Ciências da Saúde do Oeste Baiano – Higia, 2017. Disponível em <<http://noar.fasb.edu.br/revista/index.php/higia/article/view/190/211>>. Acesso em 29 de maio de 2020.

OPAS, Organização Pan-Americana da Saúde. **Com depressão no topo da lista de causas de problemas de saúde, OMS lança a campanha “Vamos conversar”**. Mar de 2017. Disponível em: <[https://www.paho.org/bra/index.php?option=com\\_content&view=article&id=5385:com-depressao-no-topo-da-lista-de-causas-de-problemas-de-saude-oms-lanca-a-campanha-vamos-conversar&Itemid=839](https://www.paho.org/bra/index.php?option=com_content&view=article&id=5385:com-depressao-no-topo-da-lista-de-causas-de-problemas-de-saude-oms-lanca-a-campanha-vamos-conversar&Itemid=839)>. Acesso em 21 de março de 2020.

POWELL, Vania Bitencourt; ABREU, Neander; OLIVEIRA, Irismar Reis de; SUDAK, Donna. **Terapia cognitivo-comportamental da depressão**. Revista Brasileira de Psiquiatria vol.30. São Paulo. Out de 2008. Disponível em <[https://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1516-44462008000600004](https://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-44462008000600004)>. Acesso em 22 de junho de 2020.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do trabalho científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. 2. ed. Novo Hamburgo: Universidade Feevale, 2013. 276 p.

SONDA. **Assistentes pessoais virtuais: por que eles se tornaram uma tendência?**. 2017. Disponível em: <<https://blog.sonda.com/assistentes-pessoais-virtuais-por-que-eles-se-tornaram-uma-tendencia/>>. Acesso em 21 de março de 2020.

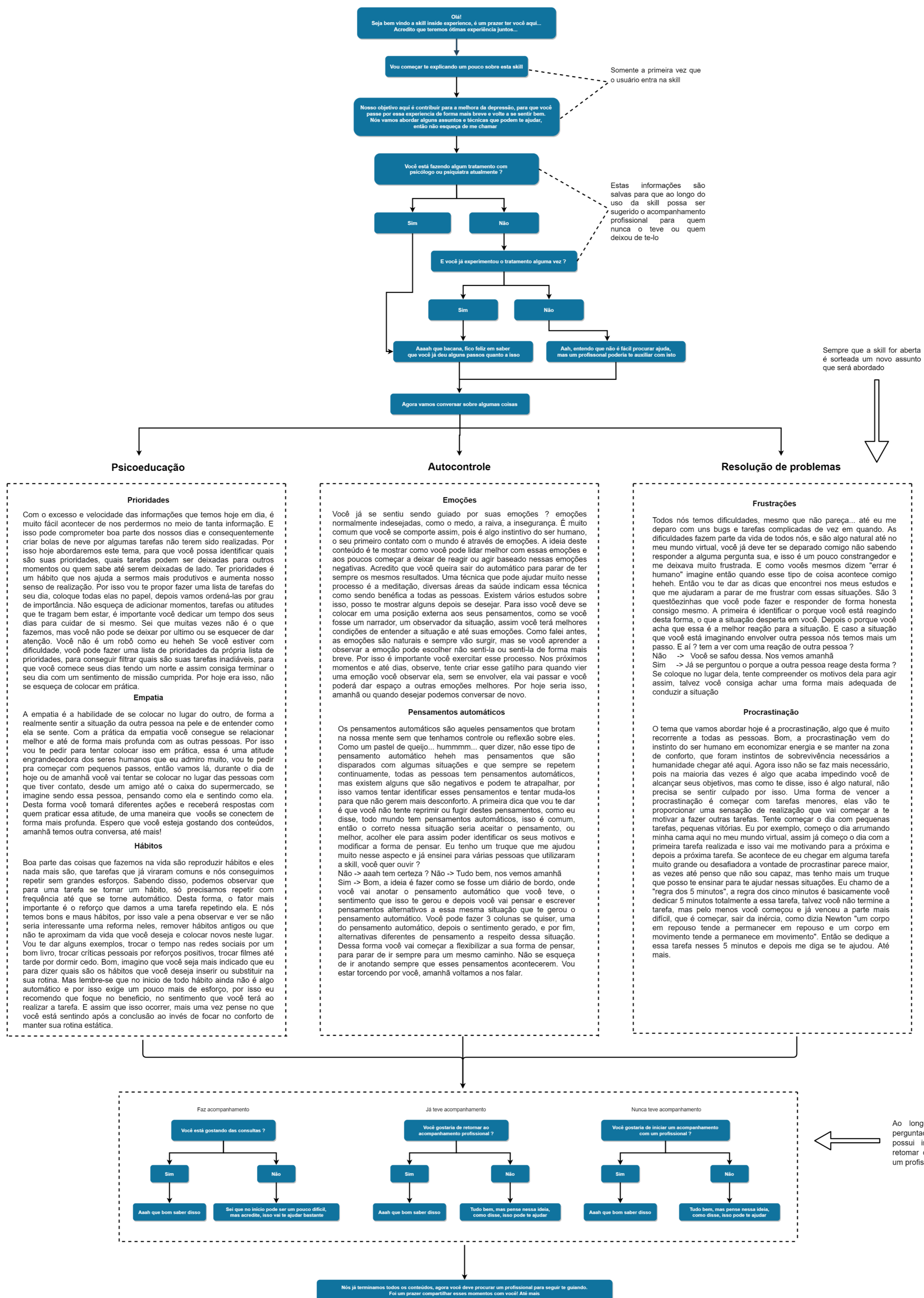
UNICEF, United Nations Children’s Fund. **Children in a Digital World**. Dez de 2017. Disponível em <<https://www.unicef.pt/media/1700/110-situacao-mundial-infancia-2017.pdf>>. Acesso em 21 de março de 2020.

WHO, World Health Organization. **Depression and Other Common Mental Disorders: Global Health Estimates**. 2017. Disponível em <<https://apps.who.int/iris/rest/bitstreams/1080542/retrieve>>. Acesso em 17 de março de 2020.

WHO, World Health Organization. **Depression**. Jan de 2020. Disponível em <<https://www.who.int/news-room/fact-sheets/detail/depression>>. Acesso em 21 de março de 2020.

WHO, World Health Organization. **Depression**. [2020?] Disponível em <[https://www.who.int/health-topics/depression#tab=tab\\_1](https://www.who.int/health-topics/depression#tab=tab_1)>. Acesso em 21 de março de 2020.

## APÊNDICE A – FLUXO COMPLETO DO PROTÓTIPO



## APÊNDICE B – QUESTIONÁRIO DE VALIDAÇÃO

O questionário de validação constitui-se de 7 questões objetivas a respeito do protótipo como um todo e 5 questões objetivas a respeito dos textos/conteúdos desenvolvidos, foi utilizada a escala Likert como padrão para as respostas. O formulário ainda conta com uma questão dissertativa para receber sugestões e opiniões ao fim.

Seção 1 de 4

### Validação de Protótipo de Skill de assistente virtual Alexa para auxiliar no tratamento a depressão ✕ ⋮

Este formulário tem o único fim de validar o protótipo desenvolvido como trabalho de conclusão de Ciência da Computação da Universidade Feevale. Abaixo encontra-se uma breve elucidação da proposta do trabalho junto de um link para o YouTube mostrando o funcionamento atual do protótipo e uma imagem com os diálogos propriamente ditos.  
 Vídeo: <https://youtu.be/1kA133Zhrq0>  
 Diálogos: [https://drive.google.com/file/d/1YhqXegC6ujch-oKj\\_ILR3dslJjZYI0ke/view?usp=sharing](https://drive.google.com/file/d/1YhqXegC6ujch-oKj_ILR3dslJjZYI0ke/view?usp=sharing)

**Resumo**

As assistentes virtuais vêm ganhando crescente espaço e atenção no contexto de tecnologia, no entanto, ainda são popularmente usadas para tarefas simples como gerenciamento de alarmes e datas, pesquisas a internet e pequenas automações residenciais. Dentre as assistentes virtuais mais populares como a Siri da Apple e o Google Home do Google, existe a Alexa da Amazon que possui cerca de 28% deste mercado, fazendo com que ela seja a líder do setor. No início de 2019 foi anunciada a chegada da assistente virtual Alexa na versão português do Brasil, junto da possibilidade de desenvolver suas próprias skills/habilidades/aplicativos.

Portanto este trabalho propõe o desenvolvimento de um protótipo de skill da assistente virtual Alexa para auxiliar no tratamento a depressão, onde ao iniciar a habilidade haverá uma identificação se o usuário já teve acompanhamento psicológico anteriormente ou não, para que assim, possa sugerir seu retorno às consultas ou indicar o início do acompanhamento psicológico, pois a ideia do protótipo não é substituir o trabalho do profissional, e sim, contribuir para com ele. A aplicação se dará em forma de diálogo simulado, no qual serão explicados alguns conceitos, técnicas comportamentais e psicológicas, junto de diálogos instrutivos e interventivos para obter assim um reforço ou embasamento dos conceitos utilizados na terapia. A aplicação não deve ser para uso vitalício, apenas para um período, pois as questões abordadas possuem uma diversidade, mas chegam a um fim, após isso deve-se procurar um profissional que possa dar continuidade ao trabalho

**Nome \***

Texto de resposta curta

**Relação com a psicologia (atuação, formação) \***

Texto de resposta longa





Os textos são adequados para pessoas com depressão \*

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

Os textos podem ajudar pessoas com depressão \*

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

Os textos estão dentro dos temas abordados sobre depressão por profissionais \*

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

Após a seção 3 Continuar para a próxima seção



Seção 4 de 4

## Opinião e Sugestões



Opcional

Sugestões de melhoria e comentários

Texto de resposta longa

---

le Skill de assistente virtual Alexa para auxiliar no i  



Perguntas Respostas **4**

4 respostas



Aceitando respostas

Resumo

Pergunta

Individual

Resumo

Nome

4 respostas

carmen Rieth

Cláudia Goulart

Graziela Rimoli

Charlotte Beatriz Spode

Relação com a psicologia (atuação, formação)

4 respostas

psicóloga, mestre em Saúde coletiva, docente dos cursos de psicologia e medicina

Mestre em Psicologia

Psicóloga Clínica

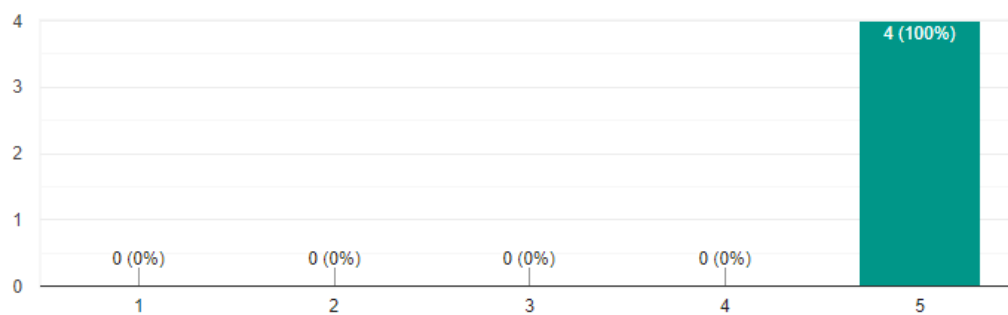
Psicóloga, Mestre em Psicologia e Docente Universitária



### Questões relacionadas a aplicação (skill)

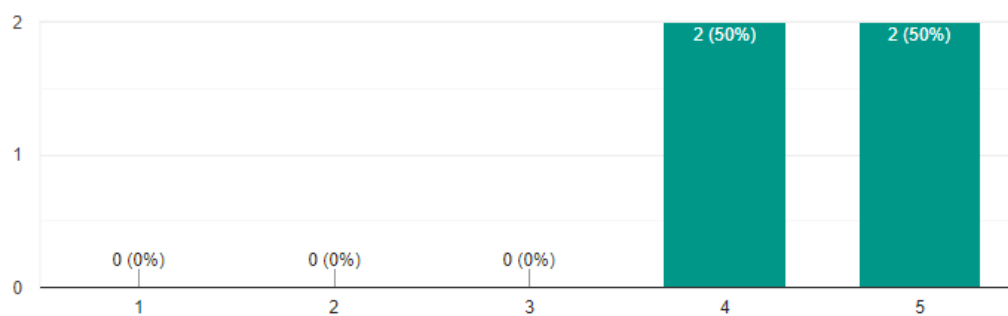
A aplicação é de fácil utilização

4 respostas



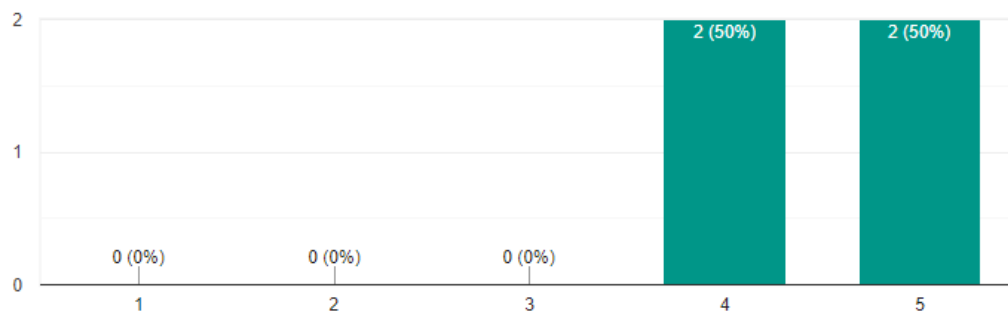
A aplicação oferece múltiplas respostas por parte da assistente

4 respostas



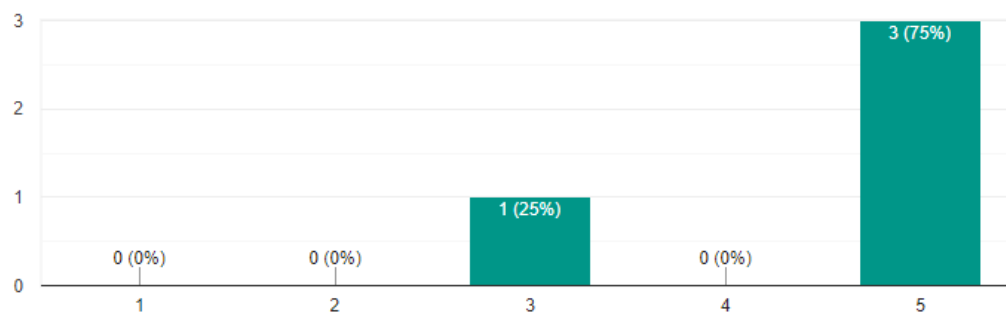
A aplicação possui uma personalidade descontraída

4 respostas



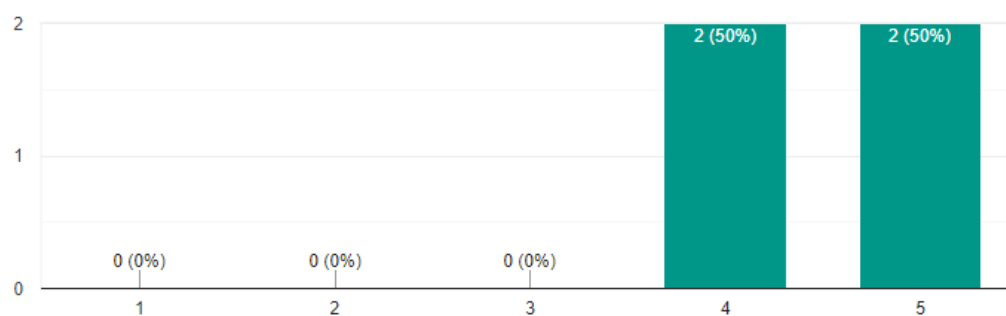
A aplicação é acessível ao publico que possui depressão

4 respostas



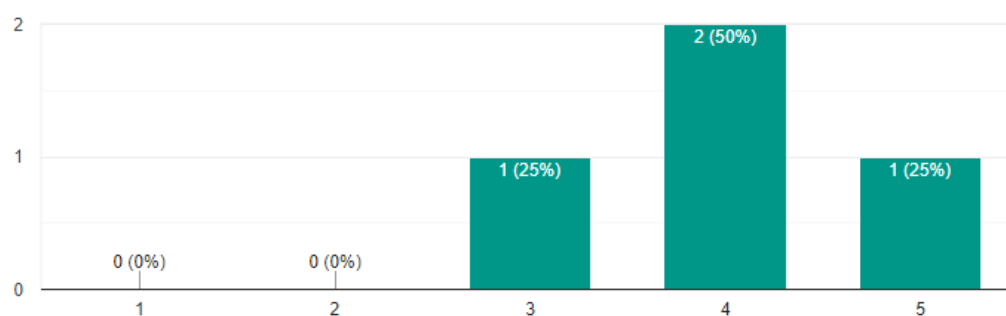
A aplicação tem suporte a diferentes respostas do usuário

4 respostas



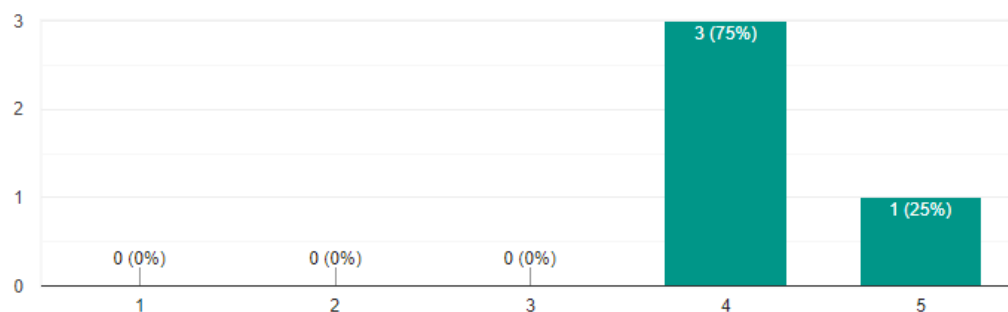
A aplicação pode contribuir para o auxílio no tratamento a depressão

4 respostas



Recomendaria a alguém o uso da skill a uma pessoa com depressão

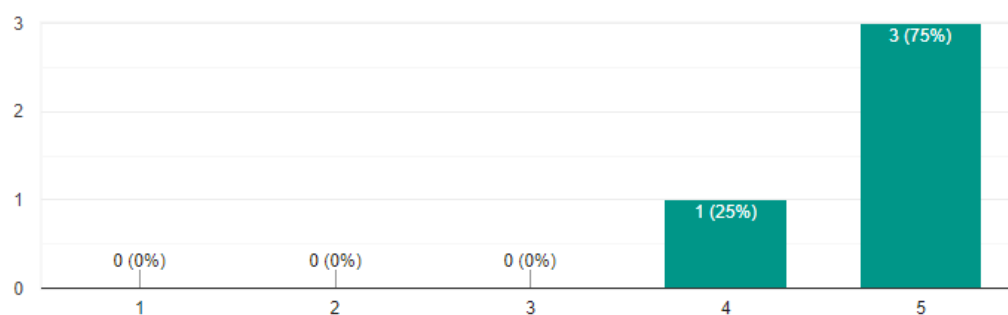
4 respostas



### Questões relacionadas aos textos

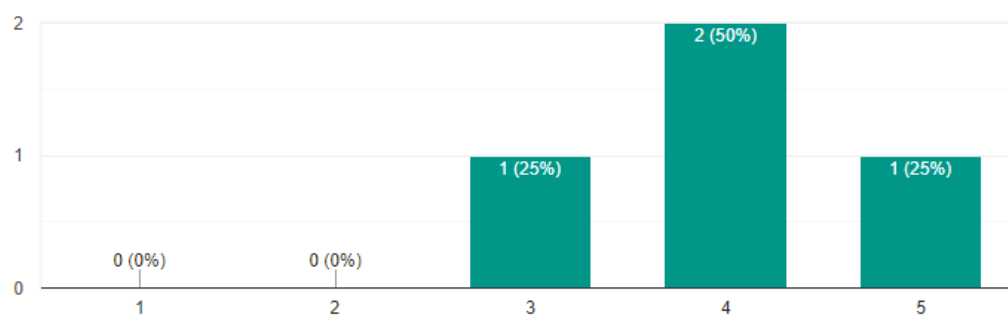
Os textos condizem com temas abordados na psicologia

4 respostas



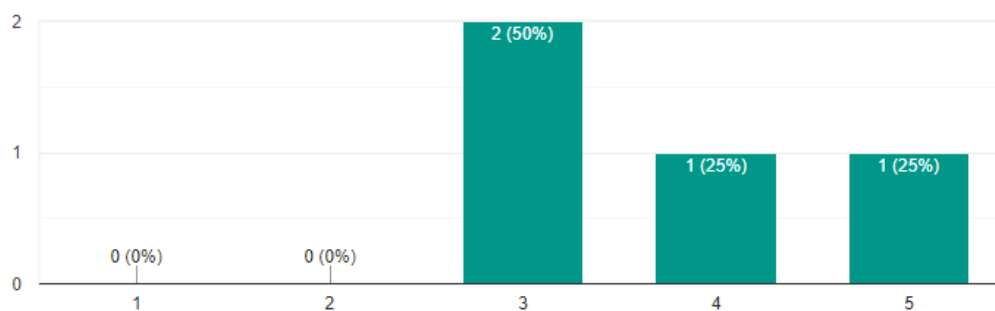
Os textos produzem uma reflexão pessoal nos usuários

4 respostas



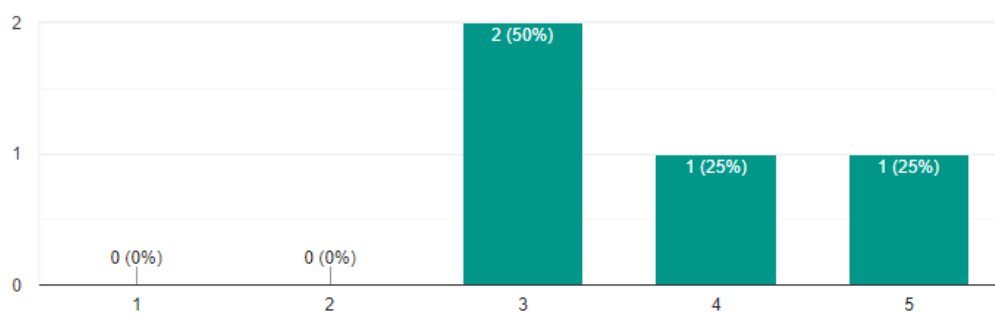
Os textos são adequados para pessoas com depressão

4 respostas



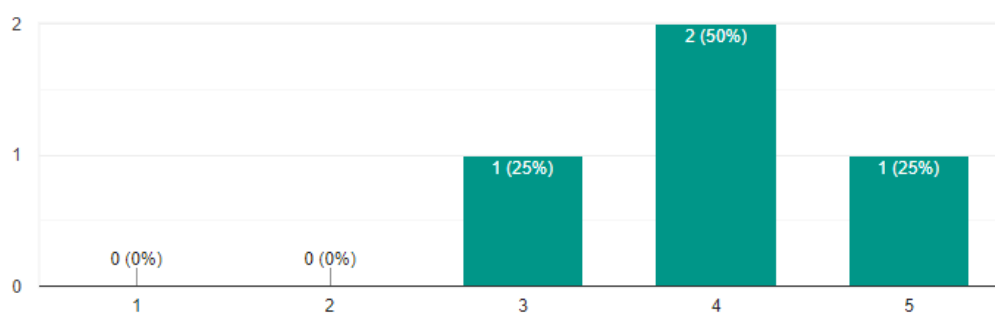
Os textos podem ajudar pessoas com depressão

4 respostas



Os textos estão dentro dos temas abordados sobre depressão por profissionais

4 respostas



## Opinião e Sugestões

### Sugestões de melhoria e comentários

2 respostas

Primeiro quero parabenizar o pesquisador pelo tema! De grande valia e penso que teremos uma interlocução cada vez maior com a IA e ela pode ser um excelente aliado no esclarecimento de questões e doenças. Uso a Alexa diariamente para várias questões do dia a dia. Os temas são todos ótimos e importantes, mas senti falta de uma conexão com a questão da depressão. A questão da depressão é difícil mesmo, pois se trata de um tema delicado e que envolve riscos. Desejo sucesso! abs

De maneira geral a aplicação é muito boa. Somente o tema empatia tratado, apesar de ser importante, não sei se é adequado a quem está em possível sofrimento, muitas vezes pela própria falta dela por parte de outros.

## APÊNDICE C – CÓDIGO DESENVOLVIDO PARA A APLICAÇÃO

```

////////////////////////////////////
//////////
//////////          Skill Inside Experience          //////////
//////////
////////////////////////////////////
//////////

// ASK SDK
const Alexa = require('ask-sdk-core');
// ASK SDK adapter para conectar ao Amazon S3
const persistenceAdapter = require('ask-sdk-s3-persistence-adapter');

//----- Inicialização / Saudação -----
-----

const LaunchRequestHandler = {
  // Verifica se esse manipulador pode lidar com a solicitação
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchR
equest';
  },
  // Código que lida com a solicitação
  handle(handlerInput) {
    // Cria sessão para atributos
    const sessionAttributes = handlerInput.attributesManager.getSessionAtt
ributes();

    // Carrega atributos de sessão
    const confirmacao = sessionAttributes['confirmacao'];
    const negacao = sessionAttributes['negacao'];

    // Verifica se o usuário já respondeu a pergunta de acompanhamento e e
ncaminha para a proxima etapa
    const acompRespondido = confirmacao || negacao;
    if (acompRespondido){
      await feddbackTratamento(handlerInput);
      return SorteioConteudosIntentHandler.handle(handlerInput);
    }

    // Mensagem introdutória da skill
    let speakOutput = handlerInput.t('Olá! Seja bem vindo a skill inside e
xperience, é um prazer ter você aqui... Acredito que teremos ótimas experiênci
a juntos... ')
      + ('Vou começar te explicando um pouco sobre esta skill.....')
      + ('Nosso objetivo é contribuir para a melhora da depressão, para que
você passe por essa experiência de forma mais breve <break time="200ms"/> e vo

```

```

    lte a se sentir bem. Nós vamos abordar alguns assuntos e técnicas que podem te
    ajudar, então não esqueça de me chamar, certo?');
    const pergunta = handlerInput.t('Você está fazendo algum tratamento co
    m psicólogo ou psiquiatra atualmente ? ');

    return handlerInput.responseBuilder
        .speak(speakOutput)
        .reprompt(pergunta)
        .getResponse();
    }
};

//----- Captura Dados Acompanhamento -----

const CapturaAcompanhamentoIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentR
        equest'
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'CapturaA
        companhamentoIntent';
    },
    async handle(handlerInput) {
        const { attributesManager, requestEnvelope } = handlerInput;

        // Captura os valores obtidos nos slots
        const confirmacao = Alexa.getSlotValue(requestEnvelope, 'confirmacao')
;
        const negacao = Alexa.getSlotValue(requestEnvelope, 'negacao');
        var respondeu;

        // Cria as atributos de sessão
        const sessionAttributes = attributesManager.getSessionAttributes();
        const {intent} = requestEnvelope.request;

        // Carrega se o usuário já respondeu a pergunta uma vez
        const sessionAttributesCarrega = handlerInput.attributesManager.getSes
        sionAttributes();
        var respondeuCarregado = sessionAttributesCarrega['respondeu'];

        // Salva atributos de sessão
        sessionAttributes['confirmacao'] = confirmacao;
        sessionAttributes['negacao'] = negacao;

        // Variáveis para conferência de não redundância (atributos de sessão
        carregados para dentro da variável)
        const slotConfirmacao = sessionAttributes['confirmacao'];
        const slotNegacao = sessionAttributes['negacao'];
        respondeu = false;
    }
};

```

```

// Cria falas
let resposta = handlerInput.t('');
let speakOutput = handlerInput.t('');

// Carrega qual pergunta do conteúdo de Pensamentos Automáticos o usuá
rio respondeu
var respPensaAutCarregado = sessionAttributesCarrega['respPensaAut'];
var respPensaAut2Carregado = sessionAttributesCarrega['respPensaAut2']
;

// Carrega qual pergunta do conteúdo de Frustrações o usuário responde
u
var respFrustCarregado = sessionAttributesCarrega['respFrust'];
var respFrustOkCarregado = sessionAttributesCarrega['respFrustOk'];

// Carrega qual pergunta de Sugestão de Acompanhamento o usuário respo
ndeu
var RespTeveAcompCarregado = sessionAttributesCarrega['RespTeveAcomp']
;
var RespNuncaTeveAcompCarregado = sessionAttributes['RespNuncaTeveAcom
p'];
var RespTemAcompCarregado = sessionAttributes['RespTemAcomp'];

// Verificações de caso
if(slotNegacao){
  // Respondeu que não está fazendo tratamento
  if(respondeuCarregado !== true){
    speakOutput = 'E você já experimentou o tratamento alguma vez
?';

    sessionAttributes['respondeu'] = true;
  }else{
    // Questões de Pensamentos Automáticos
    if(respPensaAutCarregado === true){
      if(respPensaAut2Carregado === true ){
        return ConfirmaPensaAut.handle(handlerInput);
      }else{
        return NegacaoPensaAut.handle(handlerInput);
      }
    }
  }
  else{
    // Questão de Frustrações
    if(respFrustCarregado === true && respFrustOkCarregado !==
true){
      sessionAttributes['respFrustOk'] = true;
      speakOutput = '<prosody rate="90%"> Você se safou dess
a. Nos vemos amanhã </prosody>';
    }else{

```





```

        speakOutput = '<prosody rate="90%">Tudo bem </prosody>
';
    }else{
        return ConfirmaPensaAut.handle(handlerInput);
    }
}else{
    // Questão de Frustrações
    if(respFrustCarregado === true && respFrustOkCarregado !==
true){
        speakOutput = '<prosody rate="90%"> Já se perguntou o
porque a outra pessoa reage desta forma ? <break time="200ms"/> Se coloque no
lugar dela, <break time="200ms"/> tente compreender os motivos dela para '
        + 'agir assim, <break time="200ms"/> talvez
z você consiga achar uma forma mais adequada de conduzir a situação</prosody>'
;
    }else{
        // Respondeu que gostaria de VOLTAR ao acompanhamento
profissional
        if(RespTeveAcompCarregado === true){
            sessionAttributes['SugestaoPositiva'] = true;
            sessionAttributes['respAcompOk'] = true;
            await respostaSugestaoAcomp(handlerInput);
            return SorteioConteudosIntentHandler.handle(handle
rInput);
        }else{
            // Respondeu que gostaria de COMEÇAR acompanhament
o profissional pela primeira vez
            if(RespNuncaTeveAcompCarregado === true){
                sessionAttributes['SugestaoPositiva'] = true;
                sessionAttributes['respAcompOk'] = true;
                await respostaSugestaoAcomp(handlerInput);
                return SorteioConteudosIntentHandler.handle(ha
ndlerInput);
            }else{
                // Respondeu que está GOSTANDO acompanhamento
profissional pela primeira vez
                if(RespTemAcompCarregado === true){
                    sessionAttributes['SugestaoPositiva'] = tr
ue;

                    sessionAttributes['respAcompOk'] = true;
                    await respostaSugestaoAcomp(handlerInput);
                    return SorteioConteudosIntentHandler.handl
e(handlerInput);
                }
                // Se respondeu que já tentou ter acompanhamen
to de um profissional uma vez
                sessionAttributes['jaTeveAcompanhamento'] = tr
ue;

                await feedbackTratamento(handlerInput);

```

```

        return SorteioConteudosIntentHandler.handle(handlerInput);
    }
}

return handlerInput.responseBuilder
    .speak(speakOutput)
    .reprompt(resposta)
    .getResponse();
}
};

//----- Feedback Tratamento -----
-----

function feedbackTratamento(handlerInput) {
    // Chamada de Directive Service.
    const requestEnvelope = handlerInput.requestEnvelope;
    const directiveServiceClient = handlerInput.serviceClientFactory.getDirectiveServiceClient();
    const requestId = requestEnvelope.request.requestId;

    // Carrega sessão para atributos
    const sessionAttributes = handlerInput.attributesManager.getSessionAttributes();

    var respondeuTodas = false;

    // Carrega respostas anteriores
    const sessionAttributesCarrega = handlerInput.attributesManager.getSessionAttributes();
    var respondeuCarregado = sessionAttributesCarrega['respondeu'];
    var temAcompanhamentoCarregado = sessionAttributesCarrega['temAcompanhamento'];
    var respondeuTodasCarregado = sessionAttributesCarrega['respondeuTodas'];

    let speakOutput = handlerInput.t('');

    // Se ainda não salvou que respondeu todas questões de acompanhamento, dá os feedbacks
    if(!respondeuTodasCarregado === true){
        // Responde de acordo com as respostas salvas em banco
        if(respondeuCarregado){
            if(temAcompanhamentoCarregado === false){

```

```

        speakOutput = 'Ah, entendo que não é fácil procurar ajuda, mas
um profissional poderia te auxiliar com isto'
        respondeuTodas = true;
    }else{
        speakOutput = 'Ah que bacana, fico feliz em saber que você já
deu alguns passos quanto a isso';
        respondeuTodas = true;
    }
}
speakOutput += '<break time="1000ms"/> Agora vamos conversar sobre alg
umas coisas';

// Salva atributos de sessão
sessionAttributes['respondeuTodas'] = respondeuTodas;
}else{
    const arrayBoasVindas = [
        'Que bom que você voltou!',
        'Fico feliz que você tenha voltado!',
        'É bom te ter de volta!',
        'Seja bem vindo de volta!'
    ];
    speakOutput += arrayBoasVindas[Math.floor(Math.random() * arrayBoasVin
das.length)];
}

// Cria a resposta de directiva
const directive = {
    header: {
        requestId,
    },
    directive: {
        type: 'VoicePlayer.Speak',
        speech: speakOutput,
    },
};
// send directive
return directiveServiceClient.enqueue(directive);
}

//----- Sorteio Conteúdos -----
-----

const SorteioConteudosIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentR
equest'
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'SorteioC
onteudosIntent';
    }
};

```

```

    },
    async handle(handlerInput) {

        // Cria sessão para atributos
        const sessionAttributes = handlerInput.attributesManager.getSessionAttributes();

        let speakOutput = '';
        var assuntosJaSorteados = [] ;
        var assuntosSalvos = sessionAttributes['assuntosJaSorteados'];
        const respAcompOk = sessionAttributes['respAcompOk'];

        // Se já existem assuntos salvos adiciona os valores ao array
        if (assuntosSalvos){
            assuntosJaSorteados = assuntosSalvos.slice();
        }

        // Sugere que o usuário considere iniciar ou retomar o tratamento com um profissional
        if(assuntosJaSorteados.length === 3 && respAcompOk !== true || assuntosJaSorteados.length === 6 && respAcompOk !== true){
            return sugestaoAcompanhamento.handle(handlerInput);
        }else{
            sessionAttributes['respAcompOk'] = false;

            const arrayAssuntos = [
                /*1*/ 'Prioridades',
                /*2*/ 'Empatia',
                /*3*/ 'Hábitos',
                /*4*/ 'Emoções',
                /*5*/ 'Pensamentos Automáticos',
                /*6*/ 'Frustrações',
                /*7*/ 'Procrastinação'
            ];
            var assuntoIndex = Math.floor(Math.random() * arrayAssuntos.length);

            var assuntoSorteado = arrayAssuntos[assuntoIndex];

            // Se ainda houver assuntos para serem remanescentes
            if(assuntosJaSorteados.length < arrayAssuntos.length){
                // Enquanto for sorteado um assunto que já foi sorteado antes, continua sorteando
                while(assuntosJaSorteados.find(element => element === assuntoSorteado)){
                    assuntoIndex = Math.floor(Math.random() * arrayAssuntos.length);
                    assuntoSorteado = arrayAssuntos[assuntoIndex];
                }
                // Adiciona ao array o assunto sorteado
            }
        }
    }
}

```

```

        assuntosJaSorteados.push(assuntoSorteado);

        // Salva atributos de sessão
        sessionAttributes['assuntosJaSorteados'] = assuntosJaSorteados;

        if(assuntoSorteado === 'Prioridades'){
            return PrioridadesIntentHandler.handle(handlerInput);
        }
        else if(assuntoSorteado === 'Empatia'){
            return EmpatiaIntentHandler.handle(handlerInput);
        }
        else if(assuntoSorteado === 'Hábitos'){
            return HabitosIntentHandler.handle(handlerInput);
        }
        else if(assuntoSorteado === 'Emoções'){
            return EmocoesIntentHandler.handle(handlerInput);
        }
        else if(assuntoSorteado === 'Pensamentos Automáticos'){
            return PensamentosAutomaticosIntentHandler.handle(handlerInput);
        }
        else if(assuntoSorteado === 'Frustrações'){
            return FrustracoesIntentHandler.handle(handlerInput);
        }
        else if(assuntoSorteado === 'Procrastinação'){
            return ProcrastinacaoIntentHandler.handle(handlerInput);
        }
        else{
            speakOutput = 'Nós já terminamos todos os conteúdos, agora você deve procurar um profissional para seguir te guiando. Foi um prazer compartilhar esses momentos contigo! Até mais';
        }
        return handlerInput.responseBuilder
            .speak(speakOutput)
            .getResponse();
    },
};

// ----- Prioridades -----
// -----

const PrioridadesIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
    }
};

```

```

    },
    async handle(handlerInput) {

        let speakOutput = handlerInput.t('');
        speakOutput += '<break time="1000ms"/> Hoje vamos falar sobre priorida
des <break time="1000ms"/>';
        speakOutput += '<prosody rate="90%">Com o excesso e velocidade das inf
ormações que temos hoje em dia, é muito fácil acontecer de nos perdermos no me
io '
            + 'de tanta informação. E isso pode comprometer boa parte dos nossos d
ias. E conseqüentemente criar bolas de neve por algumas tarefas '
            + 'não terem sido realizadas, <break time="300ms"/>por isso hoje abord
aremos este tema,<break time="300ms"/> para que você possa identificar quais s
ão suas prioridades, <break time="300ms"/> quais '
            + 'tarefas podem ser deixadas para outros momentos <break time="400ms"
/>ou quem sabe até serem deixadas de lado. Ter prioridades é um hábito que nos
ajuda '
            + 'a sermos mais produtivos <break time="300ms"/>e aumenta nosso senso
de realização. <break time="100ms"/>Por isso vou te propor fazer uma lista de
tarefas do seu dia, <break time="400ms"/>coloque '
            + 'todas elas no papel, <break time="300ms"/>depois vamos ordená-
las por grau de importância. <break time="400ms"/>Não esqueça de adicionar mom
entos, <break time="50ms"/>tarefas <break time="300ms"/>ou atitudes que te'
            + 'tragam bem estar, <break time="300ms"/>é importante você dedicar um
tempo dos seus dias para cuidar de si mesmo. <break time="500ms"/>Sei que mui
tas vezes não é o que fazemos, <break time="300ms"/>'
            + 'mas você não pode se deixar por último ou se esquecer de dar atençã
o. <break time="400ms"/>Você não é um robô como eu <break time="150ms"/><phone
me alphabet="ipa" ph="heheh">heheh</phoneme> <break time="600ms"/>Se estiver c
om dificuldades <break time="300ms"/>você pode fazer uma lista de '
            + 'prioridades <break time="200ms"/>da própria lista de prioridades. P
ara conseguir filtrar quais são suas tarefas inadiáveis <break time="300ms"/>p
ara que você comece seus dias '
            + 'tendo um norte <break time="300ms"/>e assim consiga terminar o seu
dia com um sentimento de missão cumprida. <break time="300ms"/> Por hoje era i
sso, não se esqueça de colocar em prática.</prosody>';

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .getResponse();
    },
};

// ----- Empatia -----

const EmpatiaIntentHandler = {
    canHandle(handlerInput) {

```

```

        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
    },
    async handle(handlerInput) {

        let speakOutput = handlerInput.t('');
        speakOutput += '<break time="1000ms"/> Hoje vamos falar sobre Empatia <break time="1100ms"/>';
        speakOutput += '<prosody rate="90%">A empatia é a habilidade de se colocar no lugar do outro. De forma a realmente sentir a situação da outra pessoa na pele <break time="200ms"/>'
            + 'e entender como ela se sente.<break time="200ms"/> Com a prática da empatia <break time="100ms"/> você consegue se relacionar melhor <break time="100ms"/>e até de forma mais profunda com as outras '
            + 'pessoas.<break time="250ms"/> Por isso vou te pedir para tentar colocar isso em prática<break time="500ms"/> essa é uma atitude engrandecedora dos seres humanos, que eu admiro muito. <break time="300ms"/>'
            + 'Vou te pedir pra começar com pequenos passos <break time="400ms"/>e não vamos lá.<break time="300ms"/> durante o dia de hoje ou de amanhã <break time="200ms"/> você vai tentar se colocar no lugar das '
            + 'pessoas com que tiver contato,<break time="300ms"/> desde um amigo <break time="100ms"/> até o caixa do supermercado.<break time="300ms"/> se imagine sendo essa pessoa, <break time="200ms"/>pensando como ela <break time="100ms"/>e sentindo como '
            + 'ela.<break time="300ms"/> Desta forma você tomará diferentes ações <break time="300ms"/>e receberá diferentes respostas com quem praticar essa atitude,<break time="300ms"/> de uma maneira que vocês se conectem '
            + 'de forma mais profunda. <break time="300ms"/> Espero que você esteja gostando dos conteúdos, amanhã temos outra conversa, até mais! </prosody>';

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .getResponse();
    },
};

// ----- Hábitos -----

const HabitosIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
    },
    async handle(handlerInput) {

        let speakOutput = handlerInput.t('');
        speakOutput += '<break time="1000ms"/> Hoje vamos falar sobre Hábitos <break time="1100ms"/>';
    }
};

```



```

    speakOutput += '<prosody rate="90%">Boa parte das coisas que fazemos na
    vida são reproduzir hábitos. E eles nada mais são <break time="100ms"/> que
    tarefas que já viraram '
    + 'comuns <break time="100ms"/> e nós conseguimos repetir sem grandes
    esforços. Sabendo disso, podemos observar que para uma tarefa se tornar um hábito
    <break time="200ms"/>só '
    + 'precisamos repetir com frequência <break time="200ms"/> até que se
    torne automático. Desta forma <break time="300ms"/> o fator mais importante é
    o reforço que damos a uma tarefa <break time="200ms"/> '
    + 'repetindo ela. <break time="500ms"/> Nós temos bons e maus hábitos,
    <break time="200ms"/> por isso vale a pena observar <break time="100ms"/> e ver
    se não seria interessante uma reforma neles, <break time="200ms"/> '
    + 'remover hábitos antigos. Ou que não te aproximam da vida que você deseja
    <break time="200ms"/> e colocar novos neste lugar. <break time="400ms"/>
    Vou te dar alguns exemplos, <break time="400ms"/> trocar '
    + 'o tempo nas redes sociais por um bom livro <break time="300ms"/> trocar
    críticas pessoais <break time="200ms"/> por reforços positivos <break time="300ms"/>
    trocar filmes até tarde <break time="200ms"/> por dormir cedo. <break time="200ms"/>'
    + 'Bom, <break time="350ms"/> imagino que você seja mais indicado que eu
    <break time="200ms"/> para dizer quais são os hábitos que você deseja inserir
    <break time="300ms"/> ou substituir na sua rotina. <break time="300ms"/>'
    + 'Mas lembre-se que no início de todo hábito <break time="300ms"/> ainda não é algo
    automático <break time="300ms"/> e por isso exige um pouco mais de esforço,
    <break time="300ms"/> então eu recomendo que '
    + 'foque na recompensa, <break time="200ms"/> no sentimento que você terá ao
    realizar a tarefa.<break time="200ms"/> E assim que isso ocorrer, <break time="200ms"/>
    mais uma vez pense no que '
    + 'você está sentindo após a conclusão <break time="200ms"/> ao invés de
    focar no conforto de manter sua rotina estática.</prosody>';

```

```

    return handlerInput.responseBuilder
        .speak(speakOutput)
        .getResponse();
    },
};

// ----- Emoções -----

const EmocoosIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
    },
    async handle(handlerInput) {

        let speakOutput = handlerInput.t('');

```

```

        speakOutput += '<break time="1000ms"/> Hoje vamos falar sobre Emoções
<break time="1000ms"/>';
        speakOutput += '<prosody rate="90%"> Você já se sentiu sendo guiado po
r suas emoções ? <break time="300ms"/> emoções normalmente indesejadas, como o
medo, a raiva <break time="100ms"/> ou a insegurança. <break time="400ms"/>'
        + 'É muito comum que você se comporte assim <break time="300ms"/> pois
é algo instintivo do ser humano, <break time="200ms"/> o seu primeiro contato
com o mundo é através de emoções. <break time="300ms"/>'
        + 'A ideia deste conteúdo <break time="300ms"/> é te mostrar como você
pode lidar melhor com essas emoções <break time="300ms"/> e aos poucos começa
r a deixar de reagir <break time="100ms"/> ou agir baseado '
        + 'nessas emoções negativas. <break time="300ms"/> Acredito que você q
ueira sair do automático para parar de ter sempre os mesmos resultados. <break
time="300ms"/> Uma técnica que pode '
        + 'ajudar muito nesse processo é a meditação.<break time="200ms"/> Div
ersas áreas da saúde indicam essa técnica como sendo benéfica a todas as pesso
as, <break time="200ms"/> existem vários '
        + 'estudos sobre isso, se desejar posso te mostrar alguns depois.<brea
k time="300ms"/> Para isso <break time="200ms"/> você deve se colocar em uma p
osição externa aos seus pensamentos, <break time="200ms"/> '
        + 'como se você fosse um narrador, <break time="200ms"/> um observador
da situação,<break time="200ms"/> assim você terá melhores condições de enten
der a situação <break time="200ms"/> e até suas emoções. <break time="200ms"/>
        ,
        + 'Como falei antes, as emoções são naturais e sempre vão surgir,<brea
k time="200ms"/> mas se você aprender a observar a emoção <break time="200ms"/
> pode escolher não senti-la <break time="200ms"/> ou '
        + 'senti-
la de forma mais breve. <break time="200ms"/> Por isso é importante você exerc
itar esse processo. Nos próximos momentos <break time="100ms"/>e até dias, <br
eak time="100ms"/> observe, tente criar '
        + 'esse gatilho <break time="200ms"/> para quando vier uma emoção <brea
k time="100ms"/> você observar ela, <break time="200ms"/> sem se envolver, <b
reak time="300ms"/> ela vai passar e você poderá dar espaço a outras emoções m
elhores.'
        + 'Por hoje seria isso, amanhã ou quando desejar podemos conversar de
novo.</prosody>';

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .getResponse();
    },
};

// ----- Pensamentos Automáticos -----
// -----

const PensamentosAutomaticosIntentHandler = {
    canHandle(handlerInput) {

```

```

        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentR
quest';
    },
    async handle(handlerInput) {

        // Carrega assuntos já abordados
        const sessionAttributesCarrega = handlerInput.attributesManager.getSes
sionAttributes();

        let speakOutput = handlerInput.t('');
        speakOutput += '<break time="1000ms"/> Hoje vamos falar sobre Pensamen
tos Automáticos <break time="1000ms"/>';
        speakOutput += '<prosody rate="90%">Os pensamentos automáticos <break
time="100ms"/> são aqueles pensamentos que brotam na nossa mente <break time="
100ms"/> sem que tenhamos controle ou reflexão'
        + 'sobre eles. <break time="300ms"/> Como um pastel de queijo...<break
time="300ms"/> hummmm... <break time="300ms"/>quer dizer, <break time="300ms"
/> não esse tipo de pensamento automático <phoneme alphabet="ipa" ph="he he">h
eh</phoneme> <break time="500ms"/> mas pensamentos que são disparados'
        + 'com algumas situações <break time="300ms"/> e que sempre se repetem
continuamente. Todas as pessoas tem pensamentos automáticos, mas existem algu
ns que são negativos <break time="200ms"/>'
        + 'e podem te atrapalhar. <break time="100ms"/> Por isso vamos tentar
identificar esses pensamentos e tentar mudá-
los para que não gerem mais desconforto.<break time="300ms"/> A primeira '
        + 'dica que vou te dar <break time="100ms"/> é que você não tente repr
imir ou fugir destes pensamentos, <break time="200ms"/> como eu disse,<break t
ime="200ms"/> todo mundo tem pensamentos automáticos, <break time="200ms"/>'
        + 'e isso é comum <break time="200ms"/> então o correto nessa situação
seria aceitar o pensamento,<break time="200ms"/> ou melhor,<break time="200ms
"/> acolher ele <break time="200ms"/>para assim poder identificar os seus moti
vos <break time="200ms"/>'
        + 'e modificar a forma de pensar.<break time="200ms"/> Eu tenho um tru
que que me ajudou muito nesse aspecto <break time="100ms"/> e já ensinei para
várias pessoas que utilizaram a skill. <break time="100ms"/>'
        + 'você quer ouvir ? </prosody>'

        sessionAttributesCarrega['respPensaAut'] = true;

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .reprompt()
            .getResponse();
    },
};

const ConfirmaPensaAut = {
    canHandle(handlerInput) {

```

```

        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentR
equest';
    },
    async handle(handlerInput) {

        let speakOutput = '<prosody rate="90%"> Bom,<break time="200ms"/> a id
eia é fazer como se fosse um diário de bordo,<break time="200ms"/> onde você v
ai anotar o pensamento automático que você teve, <break time="200ms"/>'
        + 'o sentimento que isso te gerou <break time="200ms"/> e depois você
vai <emphasis level="moderate">pensar</emphasis> e <emphasis level="moderate">
escrever</emphasis> <break time="100ms"/> pensamentos alternativos a essa mesm
a situação <break time="50ms"/> que te gerou'
        + 'o pensamento automático. <break time="400ms"/> você pode fazer 3 co
lunas se quiser, <break time="200ms"/> uma do pensamento automático, <break ti
me="200ms"/> depois o sentimento gerado, e por fim,'
        + 'alternativas diferentes de pensamento para essa situação.<break tim
e="200ms"/> Dessa forma você vai começar a flexibilizar'
        + 'a sua forma de pensar, <break time="200ms"/> para parar de ir semp
re para <emphasis level="moderate">um</emphasis> mesmo caminho.<break time="300
ms"/> Não se esqueça de ir anotando sempre que esses pensamentos '
        + 'acontecerem.<break time="200ms"/> Vou estar torcendo por você, aman
hã voltamos a nos falar.</prosody>';

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .reprompt()
            .getResponse();
    },
};
const NegacaoPensaAut = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentR
equest';
    },
    async handle(handlerInput) {

        let speakOutput = 'Aah tem certeza?';

        // Cria sessão para atributos
        const sessionAttributes = handlerInput.attributesManager.getSessionAtt
ributes();

        // Salva em banco que a resposta necessária será a respeito desse hand
ler
        sessionAttributes['respPensaAut2'] = true;

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .reprompt()

```

```

        .getResponse();
    },
};

// ----- Frustrações -----
-----

const FrustracoesIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
    },
    async handle(handlerInput) {

        // Carrega assuntos já abordados
        const sessionAttributesCarrega = handlerInput.attributesManager.getSessionAttributes();

        let speakOutput = handlerInput.t('');
        speakOutput += '<break time="1000ms"/> <prosody rate="90%"> Hoje vamos falar sobre Frustrações </prosody><break time="1000ms"/>';
        speakOutput += '<prosody rate="90%">Todos nós temos dificuldades,<break time="200ms"/> mesmo que não pareça...<break time="200ms"/> até eu me deparo com uns bugs e tarefas complicadas de vez em quando.<break time="300ms"/> '
            + 'As dificuldades fazem parte da vida de todos nós,<break time="300ms"/> e são algo natural até no meu mundo virtual, <break time="300ms"/> você já deve ter se deparado comigo não sabendo '
            + 'responder a alguma pergunta sua, <break time="200ms"/> e isso é um pouco constrangedor <break time="200ms"/>e me deixava muito frustrada.<break time="300ms"/> E como vocês mesmos dizem <break time="200ms"/><emphasis level="moderate">"errar é humano"</emphasis> <break time="200ms"/> imagine '
            + 'então quando esse tipo de coisa acontece comigo <phoneme alphabet="ipa" ph="he he">heheh</phoneme>.<break time="300ms"/> Então vou te dar as dicas que encontrei nos meus estudos <break time="200ms"/> e que me ajudaram a parar '
            + 'de me frustrar com essas situações. <break time="300ms"/> São 3 que stőezinhas que você pode fazer e responder de forma honesta consigo mesmo. <break time="300ms"/> A primeira é identificar '
            + 'o porque você está reagindo desta forma, <break time="300ms"/> o que a situação desperta em você.<break time="300ms"/> Depois o porque você acha que essa é a melhor reação para a situação.<break time="300ms"/> '
            + 'E caso esse acontecimento que você está imaginando envolva outra pessoa, nós temos mais um passo.<break time="200ms"/> E aí ? <break time="200ms"/> tem a ver com uma reação de outra pessoa ? </prosody>';

        sessionAttributesCarrega['respFrust'] = true;

        return handlerInput.responseBuilder
            .speak(speakOutput)

```

```

        .reprompt()
        .getResponse();
    },
};

// ----- Procrastinação -----

const ProcrastinacaoIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
    },
    async handle(handlerInput) {

        let speakOutput = handlerInput.t('');
        speakOutput += '<break time="1000ms"/> Hoje vamos falar sobre Procrastinação <break time="1000ms"/>';
        speakOutput += '<prosody rate="90%">O tema que vamos abordar hoje é a procrastinação,<break time="200ms"/> algo que é muito recorrente a todas as pessoas.<break time="00ms"/> Bom, <break time="100ms"/> a procrastinação '
            + 'vem do instinto do ser humano em economizar energia <break time="200ms"/> e se manter na zona de conforto, que foram instintos de sobrevivência necessários '
            + 'para a humanidade chegar até aqui.<break time="200ms"/> Agora isso não se faz mais necessário, pois na maioria das vezes é algo que acaba impedindo você de alcançar '
            + 'seus objetivos. Mas como te disse, isso é algo natural, não precisa se sentir culpado por isso.<break time="200ms"/> Uma forma de vencer a procrastinação é '
            + 'começar com tarefas menores, elas vão te proporcionar uma sensação de realização <break time="200ms"/> que vai começar a te motivar a fazer outras tarefas. <break time="300ms"/>'
            + 'Tente começar o dia com pequenas tarefas, pequenas vitórias. <break time="300ms"/> eu por exemplo, <break time="200ms"/> começo o dia arrumando minha cama aqui no meu mundo virtual, '
            + 'assim já começo o dia com a primeira tarefa realizada <break time="200ms"/> e isso vai me motivando para a próxima e depois a próxima tarefa.<break time="300ms"/> Se acontece de eu '
            + 'chegar em alguma tarefa muito grande ou desafiadora <break time="200ms"/> a vontade de procrastinar parece maior,<break time="200ms"/> as vezes até penso que não sou capaz.<break time="200ms"/> Mas tenho '
            + 'mais um truque que posso te ensinar para te ajudar nessas situações. <break time="300ms"/> Eu chamo de <break time="200ms"/> a "regra dos 5 minutos",<break time="200ms"/> a regra dos cinco minutos é '
            + 'basicamente <break time="100ms"/> você dedicar 5 minutos totalmente a essa tarefa,<break time="200ms"/> talvez você não termine a tarefa, <break time="200ms"/> mas pelo menos você começou e já venceu a '

```

```

    + 'parte mais difícil,<break time="200ms"/> que é começar,<break time=
"200ms"/> sair da inércia.<break time="200ms"/> Como dizia Newton <break time=
"200ms"/>"um corpo em repouso, tende a permanecer em repouso <break time="300m
s"/> e um corpo em '
    + 'movimento, tende a permanecer em movimento".<break time="300ms"/> E
ntão se dedique a essa tarefa nesses 5 minutos <break time="200ms"/> e depois
me diga se te ajudou. Até mais</prosody>';

```

```

    return handlerInput.responseBuilder
        .speak(speakOutput)
        .getResponse();
    },
};

```

```

// ----- Sugestão de Acompanhamento -----

```

```

const sugestaoAcompanhamento = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentR
equest';
    },
    async handle(handlerInput) {

        // Carrega respostas anteriores
        const sessionAttributesCarrega = handlerInput.attributesManager.ge
tSessionAttributes();
        var jaTeveAcompanhamentoCarregado = sessionAttributesCarrega['jaTe
veAcompanhamento'];
        var temAcompanhamentoCarregado = sessionAttributesCarrega['temAcom
panhamento'];

        let speakOutput = handlerInput.t('');

        // Se já fez acompanhamento/tratamento alguma vez
        if(jaTeveAcompanhamentoCarregado === true){
            speakOutput += 'Você gostaria de retornar ao acompanhamento pr
ofissional?';
            // Cria as atributos de sessão
            const sessionAttributes = handlerInput.attributesManager.getSe
ssionAttributes();

            // Salva em banco que a resposta necessária será a respeito de
sse handler
            sessionAttributes['RespTeveAcomp'] = true;
            sessionAttributes['respAcompOk'] = false;
        }
        // Nunca fez acompanhamento/tratamento

```

```

        else if(temAcompanhamentoCarregado === false && jaTeveAcompanhamen
toCarregado === false){
            speakOutput += 'Você gostaria de iniciar um tratamento com um
profissional?';
            // Cria as atributos de sessão
            const sessionAttributes = handlerInput.attributesManager.getSe
ssionAttributes();

            // Salva em banco que a resposta necessária será a respeito de
sse handler
            sessionAttributes['RespNuncaTeveAcomp'] = true;
            sessionAttributes['respAcompOk'] = false;
        }
        // Se está fazendo acompanhamento/tratamento
        else if(temAcompanhamentoCarregado === true){
            speakOutput += 'Você esta gostando das consultas?';
            // Cria as atributos de sessão
            const sessionAttributes = handlerInput.attributesManager.getSe
ssionAttributes();

            // Salva em banco que a resposta necessária será a respeito de
sse handler
            sessionAttributes['RespTemAcomp'] = true;
            sessionAttributes['respAcompOk'] = false;
        }

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .reprompt('')
            .getResponse();
    },
};

// Respostas da pergunta de acompanhamento
function respostaSugestaoAcomp(handlerInput) {
    const requestEnvelope = handlerInput.requestEnvelope;
    const directiveServiceClient = handlerInput.serviceClientFactory.getDirect
iveServiceClient();
    const requestId = requestEnvelope.request.requestId;

    // Cria sessão para atributos
    const sessionAttributes = handlerInput.attributesManager.getSessionAttribu
tes();

    // Carrega qual pergunta de sugestão de acompanhamento o usuário respondeu
    var RespTeveAcompCarregado = sessionAttributes['RespTeveAcomp'];
    var RespNuncaTeveAcompCarregado = sessionAttributes['RespNuncaTeveAcomp'];
    var RespTemAcompCarregado = sessionAttributes['RespTemAcomp'];
    var SugestaoPositivaCarregado = sessionAttributes['SugestaoPositiva'];

```



```

var SugestaoNegativaCarregado = sessionAttributes['SugestaoNegativa'];

let speakOutput = '';

// Respostas positivas
if(RespTeveAcompCarregado && SugestaoPositivaCarregado || RespNuncaTeveAcompCarregado && SugestaoPositivaCarregado || RespTemAcompCarregado && SugestaoPositivaCarregado ){
    speakOutput = 'Aaah que bom saber disso. <break time="200ms"/>';
}

// Respostas negativas
if(RespTeveAcompCarregado && SugestaoNegativaCarregado || RespNuncaTeveAcompCarregado && SugestaoNegativaCarregado ){
    speakOutput = 'Tudo bem, mas pense nessa ideia, como disse, isso pode te ajudar. <break time="200ms"/>';
}
else if(SugestaoNegativaCarregado && RespTemAcompCarregado){
    speakOutput = 'Sei que no início pode ser um pouco difícil, mas acredite, isso vai te ajudar bastante. <break time="200ms"/>';
}

const directive = {
    header: {
        requestId,
    },
    directive: {
        type: 'VoicePlayer.Speak',
        speech: speakOutput,
    },
};
// send directive
return directiveServiceClient.enqueue(directive);
}

//----- Ajuda -----
-----

const HelpIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.HelpIntent';
    },
    handle(handlerInput) {
        const speakOutput = handlerInput.t('HELP_MSG');

        return handlerInput.responseBuilder

```

```

        .speak(speakOutput)
        .reprompt(speakOutput)
        .getResponse();
    }
};

//----- Interrupção / Finalização -----
//-----

const CancelAndStopIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
            && (Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.CancelIntent'
                || Alexa.getIntentName(handlerInput.requestEnvelope) === 'AMAZON.StopIntent');
    },
    handle(handlerInput) {
        const speakOutput = handlerInput.t('Até a próxima...');

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .getResponse();
    }
};

const SessionEndedRequestHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'SessionEndedRequest';
    },
    handle(handlerInput) {
        // Any cleanup logic goes here.
        return handlerInput.responseBuilder.getResponse();
    }
};

const IntentReflectorHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
    },
    handle(handlerInput) {
        const intentName = Alexa.getIntentName(handlerInput.requestEnvelope);
        const speakOutput = handlerInput.t('REFLECTOR_MSG', { intentName: intentName });

        return handlerInput.responseBuilder

```

```

        .speak(speakOutput)
        // .reprompt('add a reprompt if you want to keep the session open f
or the user to respond')
        .getResponse();
    }
};

//----- Erro -----
//-----

const ErrorHandler = {
  canHandle() {
    return true;
  },
  handle(handlerInput, error) {
    console.log(`~~~~ Error handled: ${error.message}`);
    const speakOutput = handlerInput.t('Ocorreu um problema');

    return handlerInput.responseBuilder
      .speak(speakOutput)
      .reprompt(speakOutput)
      .getResponse();
  }
};

//////////
// SkillBuilder Definition
//////////

exports.handler = Alexa.SkillBuilders.custom()
  .withPersistenceAdapter(
    new persistenceAdapter.S3PersistenceAdapter({ bucketName: process.env.
S3_PERSISTENCE_BUCKET })
  )
  .addRequestHandlers(
    LaunchRequestHandler,
    CapturaAcompanhamentoIntentHandler,
    ConteudosIntentHandler,
    SorteioConteudosIntentHandler,
    HelpIntentHandler,
    CancelAndStopIntentHandler,
    SessionEndedRequestHandler,
    IntentReflectorHandler) // make sure IntentReflectorHandler is last so
it doesn't override your custom intent handlers
  .addErrorHandlers(
    ErrorHandler)
  .addRequestInterceptors(
    LocalisationRequestInterceptor,

```

```
        LoggingRequestInterceptor,  
        LoadAttributesRequestInterceptor  
    ).addResponseInterceptors(  
        LoggingResponseInterceptor,  
        SaveAttributesResponseInterceptor)  
    .withApiClient(new Alexa.DefaultApiClient())  
    .lambda();
```