



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**COMPUS DE QUIXADÁ**  
**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**RAYNARA SILVA LIMA**

**PROTÓTIPO DE UM SISTEMA DE AUTOMAÇÃO PARA O SETOR DA  
CONFEITARIA, UTILIZANDO A ASSISTENTE VIRTUAL ALEXA**

**QUIXADÁ**

**2022**

RAYNARA SILVA LIMA

PROTÓTIPO DE UM SISTEMA DE AUTOMAÇÃO PARA O SETOR DA CONFEITARIA,  
UTILIZANDO A ASSISTENTE VIRTUAL ALEXA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de computação do Compus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de computação.

Orientador: Prof. Dr. Wagner Guimarães Al-Alam.

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- L71p Lima, Raynara Silva.  
Protótipo de um sistema de automação para o setor da confeitaria, utilizando a assistente virtual Alexa / Raynara Silva Lima. – 2022.  
79 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Computação, Quixadá, 2022.  
Orientação: Prof. Dr. Wagner Guimarães Al-Alam.
1. Confeitaria. 2. Automação. 3. Assistente virtual inteligente. 4. Microempreendedor. I. Título.  
CDD 621.39
-

RAYNARA SILVA LIMA

PROTÓTIPO DE UM SISTEMA DE AUTOMAÇÃO PARA O SETOR DA CONFEITARIA,  
UTILIZANDO A ASSISTENTE VIRTUAL ALEXA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de computação do Compus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de computação.

Aprovada em: \_\_\_/\_\_\_/\_\_\_.

BANCA EXAMINADORA

---

Prof. Dr. Wagner Guimarães Al-Alam (Orientador)  
Universidade Federal do Ceará (UFC)

---

Profa. Dra. Andréia Libório Sampaio  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Cristiano Bacelar de Oliveira  
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha família, em especial aos meus pais que sempre me apoiaram e fizeram de tudo para que eu chegasse até aqui.

## AGRADECIMENTOS

Primeiramente gostaria de agradecer aos meus pais, Maria das Graças e Raimundo que sempre prezaram pela minha educação, me apoiaram, me deram forças e se esforçaram para que eu conseguisse chegar até aqui, principalmente no momento mais difícil da minha vida. A eles também agradeço pela contribuição para a escolha do tema deste trabalho, que foi pensando no trabalho deles que proporcionou meu sustento. Aos meus irmãos Myrélia e Raynan, que também sempre foram fonte de apoio, destacando minha irmã que contribuiu diretamente na realização deste trabalho. Ao meu namorado Ari, que me apoiou e me acompanhou desde o início da minha graduação.

Ao professor Wagner Guimarães Al-Alam, pela orientação, dedicação, apoio e paciência durante todo o período de realização deste trabalho.

Às minhas amigas de infância Kelly e Ana Clara pela amizade e apoio desde o início. Ao meu grupinho de amigos que fiz durante a graduação: Rafaella, Marisa, Iago, Iury e Felipe que estiveram ao meu lado, me ajudando nos momentos difíceis e dando força para continuar e não desistir. Sem vocês este período teria sido muito mais difícil. Gostaria de destacar desses amigos a Rafaella, que colaborou com este trabalho emprestando componentes necessários para o protótipo, assim como Marisa que me salvou ao me trazer alguns outros componentes que eu precisei.

Ao meu psiquiatra Dr Nadson Carvalho, que me ajudou tratando da minha depressão que me fez passar um ano e meio sem conseguir terminar este trabalho.

A todos meus professores da Universidade Federal do Ceará - Campus Quixadá, que contribuíram para a minha formação.

Às pessoas com quem convivi ao longo desses anos de curso, que me incentivaram e que certamente tiveram impacto na minha formação acadêmica.

## RESUMO

A automação veio para tornar processos manuais independentes da interferência humana, com o desenvolvimento de técnicas e equipamentos que melhoram as produções e proporcionam melhores condições de vida. Os sistemas de automação são de grande importância para pequenas empresas e para microempreendedores individuais (MEI) se tornarem mais competitivos no mercado e auxiliar em seus desafios. Dessa forma, um sistema de automação é importante para os MEIs confeitários que têm os desafios de manter o negócio visível, rentável e em atividade. Com base nisso, o objetivo deste trabalho é desenvolver um sistema de automação para auxiliar MEIs confeitários usando a assistente virtual Alexa como interface de interação, a fim de proporcionar praticidade, economia de tempo, maior higiene, além de reduzir possíveis erros. Para isso, foi realizada uma pesquisa bibliográfica, de abordagem qualitativa e exploratória com etapas de levantamento de requisitos, modelagem do sistema, descrição dos casos de uso e os diagramas de requisitos, sequência e de estados. A partir disso, foi desenvolvida uma skill Alexa, permitindo usar o sistema utilizando a voz, o que proporciona praticidade, economia de tempo já que não será preciso pegar no aparelho celular para dar os comandos, maior higiene visto que reduzirá o contato com o aparelho reduzindo assim o risco de contaminações vindas do aparelho para o alimento por meio das mãos do manipulador, além de reduzir possíveis erros. No entanto, é importante realizar um teste com usuários reais para verificar se o sistema realmente seria útil a eles, além de analisar a usabilidade, que envolve a eficácia, a eficiência e satisfação dos usuários.

**Palavras-chave:** automação; alexa; microempreendedores individuais; confeitaria; forno.

## ABSTRACT

Automation came to make manual processes independent of human interference, with the development of techniques and equipment that improve production and provide better living conditions. Automation systems are of great importance for small businesses and individual microentrepreneurs (IME) to become more competitive in the market and help with their challenges. In this way, an automation system is important for MEIs confectioners who have the challenges of keeping the business visible, profitable and active. Based on this, the objective of this work is to develop an automation system to help pastry IMEs using the Alexa virtual assistant as an interaction interface, in order to provide practicality, time savings, greater hygiene, in addition to reducing possible errors. For this, a bibliographical research was carried out, with a qualitative and exploratory approach, with requirements survey steps, system modeling, description of use cases and requirements diagrams, sequence and state machine. From this, an Alexa skill was developed, allowing the system to be used using the voice, which provides practicality, time savings since it will not be necessary to pick up the cell phone to give commands, greater hygiene since it will reduce contact with the device thus reducing the risk of contamination coming from the device to the food through the handler's hands, in addition to reducing possible errors. However, it is important to carry out a test with real users to verify if the system would really be useful to them, in addition to analyzing usability, which involves effectiveness, efficiency and user satisfaction.

**Keywords:** automation; alexa; individual microentrepreneurs; confectionery; oven.

## LISTA DE FIGURAS

Figura 1 – Arquitetura de cozinha inteligente baseada no Amazon Echo. . . . .	24
Figura 2 – Casos de uso de habilidades do Amazon Alexa para o termômetro inteligente. . . . .	24
Figura 3 – Diagrama de requisitos. . . . .	38
Figura 4 – Diagrama de sequência. . . . .	39
Figura 5 – Diagrama de estados. . . . .	40
Figura 6 – Modelo de arquitetura do sistema. . . . .	41
Figura 7 – Componentes do sistema. . . . .	43
Figura 8 – Protótipo. . . . .	43
Figura 9 – Arquitetura da comunicação. . . . .	49
Figura 10 – Trecho de código do servidor <i>web</i> com as funções de consultar agenda e agendar pedido. . . . .	78
Figura 11 – Trecho de código do <i>software</i> embarcado. . . . .	78
Figura 12 – Trecho de código do <i>software</i> com o método <i>ligaEDesligaaForno()</i> . . . . .	79
Figura 13 – Trecho de código do <i>software</i> com o método <i>enviaValorTemperatura()</i> . . . . .	79

## LISTA DE TABELAS

Tabela 1 – Tabela de comparação dos trabalhos relacionados. . . . .	27
Tabela 2 – Tabela de <i>skills</i> para a Alexa relacionadas. . . . .	27
Tabela 3 – Requisitos do sistema. . . . .	29
Tabela 4 – Caso de uso consultar agenda. . . . .	31
Tabela 5 – Caso de uso agendar pedidos. . . . .	31
Tabela 6 – Caso de uso pesquisar pedidos do cliente. . . . .	32
Tabela 7 – Caso de uso finalizar pedido. . . . .	33
Tabela 8 – Caso de uso excluir pedidos do cliente. . . . .	34
Tabela 9 – Caso de uso adicionar ingrediente no estoque. . . . .	34
Tabela 10 – Caso de uso consultar ingrediente no estoque. . . . .	35
Tabela 11 – Caso de uso excluir ingrediente do estoque em caso de consumo ou outro motivo. . . . .	35
Tabela 12 – Caso de uso ligar passagem de gás para o forno. . . . .	35
Tabela 13 – Caso de uso desligar passagem de gás para o forno. . . . .	36
Tabela 14 – Caso de uso definir tempo que produto deve ficar no forno. . . . .	36
Tabela 15 – Caso de uso consultar quanto tempo falta até a conclusão. . . . .	36
Tabela 16 – Caso de uso consultar quais são os produtos que estão no forno. . . . .	37
Tabela 17 – Caso de uso consultar a temperatura atual do forno. . . . .	37
Tabela 18 – Caso de teste para o caso de uso consultar agenda. . . . .	53
Tabela 19 – Tempos de latência de cada caso de uso. . . . .	54
Tabela 20 – Caso de teste para o caso de uso agendar pedido . . . . .	64
Tabela 21 – Caso de teste para o caso de uso excluir pedido . . . . .	66
Tabela 22 – Caso de teste para o caso de uso excluir ingrediente do estoque . . . . .	69
Tabela 23 – Caso de teste para o caso de uso pesquisar pedidos do cliente . . . . .	72
Tabela 24 – Caso de teste para o caso de uso finalizar pedido . . . . .	73
Tabela 25 – Caso de teste para o caso de uso adicionar ingrediente no estoque . . . . .	74
Tabela 26 – Caso de teste para o caso de uso ligar passagem de gás para o forno . . . . .	74
Tabela 27 – Caso de teste para o caso de uso desligar passagem de gás para o forno . . . . .	75
Tabela 28 – Caso de teste para o caso de uso definir tempo que produto deve ficar no forno . . . . .	75
Tabela 29 – Caso de teste para o caso de uso consultar quanto tempo falta até a conclusão . . . . .	76

Tabela 30 – Caso de teste para o caso de uso consultar quais são os produtos que estão no forno . . . . .	76
Tabela 31 – Caso de teste para o caso de uso consultar a temperatura atual do forno . . .	77

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	12
<b>1.1</b>	<b>Objetivos específicos</b>	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	15
<b>2.1</b>	<b>Automação</b>	15
<b>2.2</b>	<b>Assistentes Virtuais</b>	16
<b>2.2.1</b>	<i>Siri</i>	16
<b>2.2.2</b>	<i>Google Assistente</i>	17
<b>2.2.3</b>	<i>Alexa</i>	18
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	20
<b>3.1</b>	<b>Sistema de gestão para microempreendedores</b>	20
<b>3.2</b>	<b>Sistema de controle de pedidos via web padaria fácil</b>	21
<b>3.3</b>	<i>Design and implementation of smart cooking based on Amazon Echo</i>	23
<b>3.4</b>	<b>Siscont - Sistema de automatização de forno elétrico via dispositivo móvel</b>	25
<b>3.5</b>	<b>Skills Alexa relacionadas</b>	25
<b>3.5.1</b>	<i>HomeStok</i>	25
<b>3.5.2</b>	<i>Group Tasks Manager</i>	26
<b>3.5.3</b>	<i>Home Connect Oven</i>	26
<b>3.5.4</b>	<i>KitchenAid Connected Ovens</i>	26
<b>3.6</b>	<b>Relacionando os trabalhos e as skills Alexa</b>	27
<b>4</b>	<b>METODOLOGIA</b>	28
<b>4.1</b>	<b>Levantamento de requisitos</b>	28
<b>4.2</b>	<b>Modelagem do sistema</b>	29
<b>4.2.1</b>	<b>Casos de uso</b>	29
<b>4.2.1.1</b>	<i>Consultar agenda</i>	30
<b>4.2.1.2</b>	<i>Agendar pedidos</i>	31
<b>4.2.1.3</b>	<i>Pesquisar pedidos do cliente</i>	32
<b>4.2.1.4</b>	<i>Finalizar pedido</i>	33
<b>4.2.1.5</b>	<i>Excluir pedidos do cliente</i>	34
<b>4.2.1.6</b>	<i>Adicionar ingrediente no estoque</i>	34
<b>4.2.1.7</b>	<i>Consultar ingrediente no estoque</i>	35

4.2.1.8	<i>Excluir ingrediente do estoque</i>	35
4.2.1.9	<i>Ligar passagem de gás para o forno</i>	35
4.2.1.10	<i>Desligar passagem de gás para o forno</i>	36
4.2.1.11	<i>Definir tempo que produto deve ficar no forno</i>	36
4.2.1.12	<i>Consultar quanto tempo falta até a conclusão</i>	36
4.2.1.13	<i>Consultar quais são os produtos que estão no forno</i>	37
4.2.1.14	<i>Consultar a temperatura atual do forno</i>	37
4.2.2	<b>Diagrama de Requisitos</b>	37
4.2.3	<b>Diagrama de sequência</b>	38
4.2.4	<b>Diagrama de estados</b>	40
5	<b>DESENVOLVIMENTO</b>	41
5.1	<b>Desenvolvimento do sistema</b>	41
5.1.1	<i>Dispositivo controlador</i>	42
5.1.2	<i>Skill</i>	43
5.1.3	<i>Arquitetura da comunicação</i>	49
5.1.4	<i>Servidor Web</i>	50
5.1.5	<i>Banco de Dados</i>	50
5.1.6	<i>Software embarcado</i>	51
5.2	<b>Teste de software</b>	52
5.2.1	<i>Resultados e análise dos testes</i>	54
6	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	56
	<b>REFERÊNCIAS</b>	58
	<b>APÊNDICE A –TABELAS DOS CASOS DE TESTE</b>	64
	<b>APÊNDICE B – TRECHOS DE CÓDIGO</b>	78

## 1 INTRODUÇÃO

Com a evolução humana e da tecnologia, o homem busca desenvolver mecanismos para melhorar a vida e auxiliar nas tarefas do dia-a-dia (ROGGIA; FUENTES, 2016). Uma delas é a automação que veio para tornar processos manuais independentes da interferência humana, com o desenvolvimento de técnicas e equipamentos que melhoram as produções e proporcionam melhores condições de vida. (ROGGIA; FUENTES, 2016; BAYER *et al.*, 2011)

Um sistema de automação, é, portanto, um sistema de controle automático composto por sensores, atuadores, controladores e uma interface de interação humano computador (BAYER *et al.*, 2011). Exemplos desse tipo de sistema são elevadores e caixas eletrônicas que estão inseridos de forma natural no dia-a-dia das pessoas para facilitar a vida delas.

Um dos tipos de interface utilizadas em sistemas é a de voz, como alternativa a outros tipos de interação como telas, por exemplo os assistentes virtuais estão cada vez mais populares entre os usuários nas residências e no ambiente de trabalho. A ideia é fazer com a voz o que é feito com o toque, uma tendência chamada de hands free, ou "mãos livres", no qual as interações acontecem sem a necessidade de encostar no celular (OLIVEIRA, J. 2019).

Atualmente existem vários assistentes virtuais no mercado, entre eles a Siri, da Apple, o Google Assistente, da Google e a Alexa da Amazon, cada uma com suas particularidades, mas com o mesmo objetivo de auxiliar as pessoas usando a inteligência artificial.

A Alexa se diferencia dos outros assistentes pela funcionalidade de compras online e pela sua versão corporativa, o Alexa for Business. Além disso, ela possui *skills*, que são como aplicativos desenvolvidos por empresas ou indivíduos utilizando o kit de desenvolvimento chamado Alexa Skills Kit e disponibilizados na loja *Alexa Skills Store*.

O *Alexa Skill Kit* permite o desenvolvimento de *skills* de música, vídeo, notícias, de automação residencial e oferece suporte para o desenvolvimento de sistemas de automação com integração do serviço de nuvem da Amazon.

De acordo com a última edição do “Diagnóstico da automação no Brasil” promovido pela Associação Brasileira de Automação (GS1 Brasil) no ano de 2020, houve um crescimento de 3% no índice de automação no mercado brasileiro em relação a 2019 é de 7% nos últimos 4 anos. Isto mostra que as empresas estão cada vez mais buscando a automação para agilizar seus processos, facilitar sua vida e obter ganhos. Para as empresas de comércio, que vendem direto ao consumidor, a automação se amplia a todos os equipamentos que tornam os processos mais eficientes, incluindo sistemas de gestão do ponto de vendas e de estoque (GS1 BRASIL, 2017).

Esses sistemas de gestão estão englobados nos sistemas de automação comercial, que têm como objetivo, controlar, gerenciar e otimizar os processos comerciais.

Os sistemas de automação são de grande importância para pequenas empresas e para microempreendedores individuais (MEI) se tornarem mais competitivos no mercado e não sucubirem às grandes empresas que têm o acesso mais fácil às tecnologias devido ao seu maior capital. (GUERRA; TEIXEIRA, 2010) MEIs são profissionais autônomos regularizados como pequenos empresários que não sejam sócio, administrador ou titular de outra empresa, possuem no máximo um funcionário e faturamento anual de R\$81.000 ou R\$6.750 por mês. (SEBRAE, 2022b)

Segundo o Portal do Empreendedor, no Brasil existem mais de 10 milhões de MEIs e juntamente com as pequenas empresas, possuem um papel importante para a economia brasileira sendo responsáveis por cerca 30% do PIB brasileiro, além de pelo menos 1,7 milhão de famílias dependem exclusivamente da renda do microempreendedor individual.

De acordo com um estudo do SEBRAE (Serviço Brasileiro de Apoio às Micro e Pequenas Empresas) no cenário atual a pandemia do COVID-19 fez com que crescesse muito o número de negócios de alimentos e bebidas administrados por MEIs. (SEBRAE, c2022). Dentro do setor de alimentos, existe a confeitaria, que junto do mercado de doces chega a faturar 12 bilhões de reais por ano (MAYER, 2021). Além disso, segundo a pesquisa ‘Consumo Equilibrado: uma nova percepção sobre o açúcar’, realizada pelo Instituto Dante Pazzanese, grande parte dos brasileiros preferem os bolos e biscoitos ao invés do chocolate em barra e de pacotes de bombom. (MAYER, 2021)

Porém, assim como todo empresário, os MEIs confeitários têm o desafio de manter o negócio visível, rentável e em atividade. A pesquisa “Perfil do microempreendedor individual de alimentos e bebidas no Brasil: indicadores, desafios e tendências”(SEBRAE *et al.*, 2017) realizada pelo SEBRAE, ABIP (Associação Brasileira da Indústria de Panificação e Confeitaria) E ITPC (Instituto Tecnológico da Panificação e Confeitaria), revela que as principais dificuldades estão ligadas à rotina e estrutura física de equipamentos e matéria-prima, sendo citada também as operações administrativas. Continuando a pesquisa, houve um indicativo que majoritariamente esses empreendedores possuem até cinco equipamentos, com predominância dos básicos, como fogão, forno, freezer, liquidificador e batedeira.

Baseado na importância dos MEIs, da confeitaria, do uso de sistemas de automação e nas dificuldades dos MEIs confeitários, o objetivo deste trabalho é desenvolver um sistema de

automação para auxiliar MEIs confeitheiros usando a assistente virtual Alexa como interface de interação, a fim de proporcionar praticidade, economia de tempo, maior higiene, além de reduzir possíveis erros.

O sistema integra elementos da automação comercial, com o gerenciamento de agenda e controle de estoque, aliados à automação de forno a gás com temporizador, luz para visualizar melhor o conteúdo dentro dele, a possibilidade de liberar ou bloquear a passagem de gás e saber a temperatura atual perguntando à Alexa.

### **1.1 Objetivos específicos**

- Especificar as funcionalidades do sistema que envolvem o gerenciamento de agenda, de estoque e automação do forno.
- Desenvolver o sistema usando a interface de voz da Alexa, tornando-o capaz de realizar as funcionalidades especificadas.
- Verificar se o sistema funciona corretamente dentro de um tempo aceitável e cumpre o objetivo, para então ser validado.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção dedica-se a apresentar conceitos utilizados ao longo do trabalho, começando com a automação, apresentando o que é e onde pode ser aplicada. Depois, são apresentadas assistentes virtuais, contextualizando o que são elas e apresentando três das principais assistentes virtuais do mercado.

### 2.1 Automação

A palavra automação vem do latim *automatus*, que significa “mover-se por si”, isto é, realizar ações de forma automática, sem a interferência humana (BAYER; ECKHARDT; MACHADO, 2011), portanto, está ligada a utilização de sistemas automáticos para executar comandos, obter medidas, regular parâmetros e controlar funções. (SEBRAE, c2022) A automação também pode ser vista como a integração entre um ou mais sistemas de controle de um equipamento, com o objetivo de aumentar a eficiência de um determinado processo e proporcionar maior conforto, informação e segurança (SEBRAE, c2022).

Desde a pré-história o homem busca desenvolver mecanismos e invenções que reduzam o esforço físico e auxiliem nas atividades do dia-a-dia, como por exemplo, a invenção da roda. Entretanto, a automação só obteve destaque na metade do século XVIII, e até o início do século XX os dispositivos eram simples e semi-automáticos (BAYER *et al.*, 2011)). No início do século XX, surgiram os sistemas inteiramente automáticos quando a tecnologia da automação passou a contar com computadores, servomecanismos e controladores programáveis, sendo os computadores a base da tecnologia da automação contemporânea (BAYER *et al.*, 2011).

A automação pode ser aplicada em praticamente todas as áreas e classificadas de acordo com elas. Dentre as classificações, tem-se: a automação comercial, industrial, residencial, bancária, agrícola, de comunicações, de transportes, etc. Na área empresarial os tipos de automação utilizadas são a comercial e a industrial.

A automação comercial é a utilização de ferramentas e métodos tecnológicos para automatizar atividades de rotina e que normalmente são feitas manualmente em uma empresa (SEBRAE (ES), 2019). Essas atividades são relacionadas a processos comerciais, como gestão administrativa e em alguns casos a hardwares, como troca ou instalação de impressoras.

Já a automação industrial automatiza máquinas e equipamentos para transferir o processo de produção manual para um conjunto de elementos tecnológicos. Esses elementos

tecnológicos são os operacionais (sensores, motores, válvulas) e de controle, que é a parte programável. (SILVEIRA, c2020)

Portanto, o uso de automação nas empresas proporciona um aumento na eficiência, produtividade e na qualidade do produto, além da redução de custos, tornando-a mais competitiva no mercado (ARAÚJO JÚNIOR *et al.*, 2003).

## 2.2 Assistentes Virtuais

Os assistentes virtuais são softwares que permitem a interação entre um ser humano e um computador por meio de comandos de voz (RODRIGUES, 2019). Geralmente estão associadas a algum mecanismo de inteligência artificial que os tornam softwares inteligentes capazes de detectar estímulos externos, respondê-los adequadamente (MACEDO *et al.*, 2007). Diferente dos softwares tradicionais, assistentes virtuais agem de forma autônoma, adaptativa, podendo ter seu comportamento alterado de acordo com comandos do usuário e utilizam a tecnologia de reconhecimento de voz (CHIARADIA *et al.*, 2017).

O objetivo dos assistentes virtuais em geral é auxiliar as pessoas em tarefas do dia-a-dia, como organização de agenda, anotações, ajuste de despertadores e controle de dispositivos inteligentes tornando-se humanizados podendo adotar voz própria bem como opinião, humor e personalidade (MAUÉS, 2019).

Alguns dos principais assistentes no mercado são a Siri da Apple, o Google Assistente do Google e a Alexa da Amazon, todos eles possuem seus aplicativos e *Smart Speakers*, alto-falantes inteligentes que permitem escutar, interpretar e responder a um usuário via Wi-Fi (BICUDO, 2019).

### 2.2.1 Siri

A Siri é uma assistente virtual conversacional da Apple pioneira no mercado de assistentes virtuais lançada em 2011 com o iPhone 4s. Presente em todos os dispositivos da marca e no *smart speaker* HomePod, a Siri é ativada falando “Hey Siri” ou apenas “Siri” e realiza várias tarefas por comandos de voz, sem que usuário diga palavras predeterminadas ou comandos específicos com meio de interação mais amigável (SOUZA, 2016).

Entre as tarefas realizadas por ela estão: fazer chamadas ou enviar mensagens de texto, anunciar mensagens nos AirPods, definir alarmes, temporizadores, lembretes, procurar e

sugerir músicas e entre outras funções básicas de uma assistente (APPLE, c2020). Além dessas funcionalidades, com o app Home, ela é capaz de controlar aparelhos inteligentes para, por exemplo, abrir a porta da garagem, destrancar a porta da frente e acender as luzes (APPLE, c2020).

Segundo a Apple (c2020), a Siri se torna cada vez mais inteligente com o aprendizado de máquina, além de ter sido projetada para aprender offline diretamente no dispositivo. Atualmente é possível personalizá-la escolhendo o idioma entre os 21 disponíveis e ensinando quem são os familiares (APPLE, c2020).

Quanto à segurança e privacidade, de acordo com a Apple (c2020), as informações pessoais não são coletadas para vender a anunciantes ou outras organizações e a criptografia de ponta a ponta garante que os dados sincronizados entre seus dispositivos não possam ser acessados, mesmo pela Apple.

Além das funcionalidades já disponíveis, o SiriKit permite que os desenvolvedores criem aplicativos iOS, iPadOS e watchOS que funcionem com a Siri. De acordo com Turton (2017), o SiriKit funciona usando um conjunto de domínios, que representam áreas de funcionalidade relacionadas. Esses domínios estão limitados em atividades físicas, lista e notas, pagamentos, mensagens, gerenciamento de fotos, reservas de viagens, integração de veículos e códigos visuais (THAKUR, 2019).

### **2.2.2 Google Assistente**

O Google Assistente é o assistente virtual desenvolvido pela Google com o propósito de facilitar as tarefas do dia-a-dia e tornar a vida dos usuários mais simples, dando velocidade e facilitando interações (MOL, 2020). No Brasil, ela foi lançada em 2017 e hoje está disponível em 31 idiomas.

De acordo com o Google (c2020), as tarefas realizadas por ele estão divididas entre gerenciamento de tarefas, organização do dia, criação de lembranças, recebimento de respostas, controle da casa, controlando aparelhos domésticos inteligentes e diversão com o controle do Google Play e o YouTube Music. No Brasil essas tarefas estão disponíveis tanto em smartphones android e iPhone como no alto-falante, o Google Home e para ativá-lo, basta dizer “Ei, Google” ou “Ok, Google”.

Além de estar disponível nos *smartphones* e no Google Home, ele também está em alguns modelos de Android Tv com as tarefas de organização do dia, controle de casa, descobrir

novos programas e pedir para assisti-los. Em alguns idiomas ele pode está mais dispositivos como visores inteligentes, carros, *laptops*, *tablets*.

No quesito reconhecimento de voz, foi criado o *Voice Match* que permite a identificação de voz e na segurança, o usuário pode compartilhar apenas aquilo que desejar como atividade da Web e de aplicações, informações das aplicações dos dispositivos, informações dos contatos dos dispositivos, e gravações de áudio. Se ativadas, a Google utiliza os dados para tornar o assistente mais útil, aprendendo com as atividades passadas e melhorando os anúncios.

O Google Assistente também pode ser integrado com outros aplicativos como Spotify, WhatsApp, Netflix e outros aplicativos também podem ser integrados por desenvolvedores com o Actions on Google, a plataforma para desenvolvedores do Google Assistente.

As Actions são o ponto inicial da interação do usuário e do Google Assistente, utilizando a linguagem natural no qual o usuário pode falar ou digitar a frase, informando o nome da *action* que deseja interagir. Essas *actions* podem ser nas categorias de casa inteligente, pedidos de comida, jogos e ações personalizadas.

### 2.2.3 *Alexa*

Lançada em 2014, a Alexa é a assistente virtual baseada em nuvem e com reconhecimento de voz da Amazon disponível em vários dispositivos da Amazon como a família de dispositivos Amazon Echo, FireTV, Fire Tablet e de fabricantes de dispositivos terceiros (AMAZON ALEXA, c2020). A versão em português chegou no Brasil apenas em outubro de 2019, porém com o diferencial de identificar diferentes sotaques e compreender gírias e expressões regionais. Criada com foco em auxiliar as pessoas nas tarefas do dia a dia, a Alexa é capaz de executar tarefas, configurar alarmes e lembretes, reproduzir músicas e até mesmo controlar dispositivos domésticos inteligentes (GOGONI, c2020).

Juntamente com a Alexa, foi lançado o Amazon Echo, uma caixa de som inteligente, sem fio e controlado por voz com a mesma (GOGONI, c2020). Atualmente, já existem alguns modelos de Echo, incluindo o com tela, além do aplicativo utilizado em celulares e *tablets*.

Para utilizar a Alexa, primeiramente é necessário “acordá-la” com uma palavra de ativação que pode ser configurada para “Alexa”, “Amazon” ou “Echo” e só depois pedir o comando (GOGONI, c2020). Essa palavra é capturada pelo microfone do dispositivo que está sempre atento, esperando-a para depois executar o pedido sem precisar utilizar as mãos, trazendo o conceito de “mãos livres”.

A Alexa já possui funcionalidades nativas, porém ela permite criar *skills*. O desenvolvimento delas é realizado com o *Alexa Skills Kit (ASK)* que fornece APIs e ferramentas para a criação de *skills* de casa inteligente, conteúdo e notícias, educação, vídeo, música ou personalizadas (AMAZON ALEXA, c2020).

Quanto à privacidade, a Alexa grava apenas as conversas quando é detectada a palavra de ativação e as gravações podem ser revistas e excluídas. Além disso, é possível gerenciar as informações do histórico do estado dos dispositivos de casa inteligente e as permissões das *skills*.

Além de desenvolver *skills*, é possível construir produtos integrados a Alexa embutida, conectar dispositivos a ela e criar soluções corporativas. As soluções corporativas são criadas com o *Alexa for Business*, habilitando aplicativos de negócios por voz, para que os funcionários participem de reuniões, obtenham informações e relatem problemas apenas perguntando a Alexa (AMAZON ALEXA, c2020).

Nas soluções corporativas também tem o *Alexa for Hospitality* que segundo a Amazon Alexa (c2020) simplifica tarefas para os hóspedes, como tocar música, ligar para a recepção, solicitar serviço de quarto, controlar a temperatura ou a iluminação do quarto e fazer o *check-out* para aumentar as receitas, reduzir custos e melhorar a satisfação do hóspede.

Os produtos integrados da Alexa são dispositivos com a Alexa embutida, com microfone e alto-falante criados com o *Alexa Voice Service (AVS)*, que permite o acesso aos recursos da Alexa, simplificando construção de dispositivos de encaminhamento de voz e possuem microfone e alto-falante (AMAZON ALEXA, c2020). Esses produtos funcionam com as *skills* da Alexa e dispositivos inteligentes compatíveis com a mesma.

É possível também conectar os dispositivos com a Alexa para controlar dispositivos de casa inteligente com a voz ou automaticamente. Conectando com a Alexa, não é preciso projetar uma interface de usuário de voz (VUI), basta usar os nativos da Alexa, os usuários podem controlar os seus dispositivos de qualquer lugar com o Alexa embutido e pode-se integrar os dispositivos com Alexa de várias formas.

### 3 TRABALHOS RELACIONADOS

Nesta seção são apresentados trabalhos e skills que se relacionam com este nos aspectos de automação comercial ou de automação de equipamentos de cozinha que podem ou não utilizar assistentes virtuais.

#### 3.1 Sistema de gestão para microempreendedores

No trabalho do Barros (2015) é relatado o desenvolvimento do SGPME (Sistema de Gestão para Microempreendedores), um sistema web para auxiliar a organização de microempresas com o objetivo de afetar no mínimo quatro áreas importantes para o negócio, que são o atendimento ao cliente, vendas, compras e tomada de decisão.

As necessidades para os participantes indispensáveis do projeto, os stakeholders e usuários desse sistema são gerenciar materiais em estoque, pedidos pendentes e atendidos, analisar o perfil dos clientes através das vendas, analisar as informações financeiras e obter informações estratégicas sobre os produtos.

Esse sistema pretende abranger empreendedores individuais e microempresas que necessitam de um sistema de gestão eficiente, com as características de ser plataforma web open source com baixo custo de manutenção, fácil utilização e customizado para empreendedores individuais e micro empresas.

Para desenvolvê-lo, o autor lista os requisitos do sistema, planeja o projeto e modela o sistema. Entre os requisitos do sistema estão:

- Cadastrar, alterar e pesquisar fornecedor;
- Cadastrar, alterar e pesquisar produto, com o produto contendo apenas a descrição;
- Cadastrar, alterar e pesquisar cliente;
- Cadastrar compra, com fornecedor, nota/recibo, produto, descrição do produto, quantidade e valor unitário;
- Cadastrar, alterar e pesquisar pedido de venda, com pedido de venda contendo número do pedido, cliente, produto, descrição do produto, quantidade no estoque, valor de venda por item, quantidade da venda por item, valor total da venda por item, valor total da venda, data de previsão de entrega e quantidade entregue;
- Cadastrar, alterar, pesquisar e excluir catálogo;

- Emissão de relatórios;
- Visão de negócio;

O projeto utiliza UML (*Unified Modeling Language*) para ilustrar os processos necessários no desenvolvimento, apresentando diagramas de casos de uso, de classes, sequência e atividades. Como é um sistema web, as tecnologias utilizadas para desenvolvimento foram a plataforma java, o *framework* VRaptor3, o *Bootstrap* para facilitar o desenvolvimento de componentes de interface e o MySQL<sup>1</sup> para gerenciar o banco de dados.

Os primeiros resultados obtidos foram negativos por causa da indefinição da tecnologia e depois o autor ainda precisou aprender algumas tecnologias que utilizaria, mas no final desenvolveu um projeto com codificação organizada, de fácil manutenção, bem documentada e com algumas tecnologias que auxiliam no desenvolvimento e na usabilidade do usuário. O sistema desenvolvido pode ser utilizado em um micro computador ou smartphone e como projetos futuros é recomendado a continuação do desenvolvimento do sistema, já que não foram desenvolvidas todas as funcionalidades do sistema.

Este é um projeto com um bom planejamento, utilizando os diagramas da UML e possui muitas funcionalidades necessárias para um microempreendedor, porém essas funcionalidades são genéricas e não podem ser utilizadas em alguns tipos de negócios, como o deste trabalho, o da confeitaria.

### **3.2 Sistema de controle de pedidos via web padaria fácil**

Batista (2015) propõe um projeto de sistema web de controle para a padaria de um Supermercado, lidando com o setor de produção, provendo funções que possibilitem ajudar no controle de pedidos. O objetivo é desenvolver um sistema que proporciona agilidade e segurança nas informações da movimentação diária do setor de produção.

Primeiramente, Batista (2015) faz um levantamento das necessidades da padaria, buscando obter informações que serão fundamentais para a elaboração do projeto e depois é que o software é implementado com base na documentação elaborada no levantamento.

Na documentação há o levantamento e análise de requisitos, diagramas de casos de uso, diagrama de classe e de atividades da UML. Os requisitos levantados foram: efetuar login, cadastro de funcionários, cadastro de fornecedores, cadastro de clientes, cadastro de produtos com descrição, o preço e o estoque do produto, manter compras, estoque e produção.

---

<sup>1</sup> <https://www.mysql.com/>

Os diagramas de caso de uso são:

- Efetuar login;
- Manter funcionários, que permite que o gerente cadastre, altere, pesquise e exclua funcionários;
- Manter fornecedores, que permite que o gerente cadastre, altere, pesquise e exclua fornecedores;
- Manter clientes, que permite que o gerente cadastre, altere, pesquise e exclua clientes;
- Manter produtos, que permite que o gerente cadastre produtos com descrição, preço e estoque do produto, altere, pesquise e exclua produtos;
- Manter pedidos, que permite que o gerente cadastre, altere, pesquise e exclua pedidos, com os pedidos contendo produtos;
- Solicitar encomendas, que permite que o gerente/funcionário cadastre, altere, pesquise e exclua encomendas, com encomendas contendo produtos, data e quantidade;
- Manter produção que permite que gerente/funcionário possa liberar produtos para a linha de produção;

Para a implementação do sistema, foi utilizado o ambiente de desenvolvimento *Visual Studio 2012 Ultimate*, com a linguagem de programação C. Ao final o autor concluiu que o sistema atende as necessidades dos funcionários e auxilia no controle de fluxo de uma padaria, permitindo um melhor acesso às informações e movimentações do estabelecimento. O sistema atende todo o ramo de padarias e lojas que vendem produtos alimentícios, como restaurantes, pizzaria, etc, permitindo um planejamento da produção dos mesmos.

Para trabalhos futuros o autor propõe a continuação do sistema para aprimorá-lo e ainda desenvolvê-lo para dispositivos móveis para que o funcionário possa ter em mãos um dispositivo móvel com o seu sistema e assim efetuar o seu trabalho na rua ou em qualquer lugar.

O sistema é útil para o público a que se propõe, com funcionalidades importantes, porém não tem muita praticidade e mobilidade no uso, visto que é um sistema web acessado apenas em um computador e não possui nem uma versão para dispositivo móvel, tampouco com um assistente de voz.

### 3.3 *Design and implementation of smart cooking based on Amazon Echo*

O trabalho de Xiaoguang et al. (2018) trata-se do projeto e implementação de uma cozinha inteligente baseada no dispositivo Amazon Echo. O objetivo é auxiliar no cozimento dos alimentos utilizando um termômetro inteligente que fornece informações sobre a temperatura da comida, define temperatura alvo, programa alarme, prevê e mostra dinamicamente o tempo de conclusão dos alimentos para cozimento específico e programa.

O Amazon Echo é o hardware principal, ele captará o que uma pessoa diz, que pode ser por exemplo “qual é a temperatura atual da minha comida?” e também falará frases de resultado das solicitações como “a temperatura atual da sua comida é de 120 graus Fahrenheit”.

A imagem abaixo apresenta a arquitetura da cozinha inteligente, Amazon Alexa Voice Service receberá as informações que o Amazon Echo enviar através do wi-fi, reconhecerá pelo programa de reconhecimento de fala interno e então uma das intenções pré desenhadas deve ser correspondida de acordo com as palavras-chave identificadas.

O AWS Lambda executa o código escrito em Node.js com parâmetros passados do Amazon Alexa e chama os serviços de nuvem personalizados remotamente quando precisar de informações necessárias. E então, o AWS Lambda envia de volta as sentenças de resultado para o Amazon Alexa após a conclusão da execução.

Os serviços em nuvem mantêm as conexões de utensílios de cozinha inteligentes e reúnem as informações de condição em banco de dados incorporado em tempo real. Os serviços em nuvem encapsulam as informações consultadas do banco de dados no formato JSON e transmitem os dados JSON para o AWS Lambda. Os serviços em nuvem podem conectar diretamente o wi-fi equipado com utensílios de cozinha inteligentes e conectar indiretamente os outros utensílios de cozinha equipados com *Bluetooth* por meio de aplicativos acessórios.

Figura 1 – Arquitetura de cozinha inteligente baseada no Amazon Echo.



Fonte: Fonte: Xiaoguang et al., 2018.

Na Figura 1 são mostrados os casos de uso do Amazon Alexa Skill para o termômetro inteligente, como consultar a temperatura atual do cozimento dos alimentos e definir a temperatura-alvo do cozimento. Esses casos de uso devem ser bem projetados com frases reconhecidas pelo Amazon Alexa e frases esperadas correspondentes faladas pelo Amazon Echo, como mostra a Figura 2.

Figura 2 – Casos de uso de habilidades do Amazon Alexa para o termômetro inteligente.

Intent	Sentences	Expected Reply
Query Current Temperature	What's the temperature of my food?	Your food is currently at ** degrees Fahrenheit.
	How hot is my food?	
	What's the current temperature of my food?	
Set Target Temperature	Set the target temperature to ** degrees.	Ok, your Target Temperature has been set to ** degrees.
	Set thermometer to ** degrees.	
	Tell thermometer to set the temperature alarm for ** degrees.	
Query Completion Time	When will my food be ready?	Your thermometer predicts that the time-to-temperature is xxx minutes.
	How long until my food is done?	
	Is my food ready?	
Set alarm	Notify me when my food is done.	Ok, your temperature alarm will ring unit as well
	Set an alarm for when my food is ** degrees.	
	Let me know when my food is ** degrees.	

Fonte: Fonte: Xiaoguang et al., 2018.

Portanto, este projeto apresenta um sistema com base no Amazon Echo, o dispositivo que utiliza a assistente virtual Alexa para uma cozinha inteligente, mas focado em um termômetro, diferente deste trabalho que é focado em um forno.

### **3.4 Siscont - Sistema de automatização de forno elétrico via dispositivo móvel**

O artigo de Pereira (2017) apresenta um exemplo de eletrodoméstico automatizado, no caso um forno elétrico doméstico, controlado por um dispositivo móvel. O objetivo dele é proporcionar comodidade aos usuários e propiciar agilidade em suas tarefas diárias.

O sistema utiliza comunicação sem fio entre o Sistema de Controle do Forno (SCF) e o aplicativo mobile para assim ser possível controlar a temperatura e agir como uma chave liga/desliga.

Nesse sistema o usuário pode selecionar uma temperatura através do aplicativo que é enviada para o SCF e assim acionar o forno e estabelecer a temperatura indicada. O usuário também pode alterar a temperatura durante o funcionamento e também interromper ele utilizando o aplicativo ou de forma manual.

Um dos componentes utilizados é o sensor de temperatura, o termopar tipo k, um sensor simples, robusto e de baixo custo. Ele tem como função no sistema gerar uma Força Eletro-Motriz, que após compensação do MAX6675 é enviada via SPI com o sinal já tratado ao microcontrolador e mostrada em um display LCD.

Após a conclusão do desenvolvimento do sistema, o autor identificou que os objetivos propostos foram atingidos, sendo eles: maior facilidade no preparo de alimentos, controle sem precisar manter contato visual com o forno e facilidade de entendimento e manuseio do sistema. A principal diferença dos dois trabalhos se dá pelo tipo de forno automatizado que este é um elétrico, enquanto o outro é um forno a gás, porém nos dois são utilizados o sensor de temperatura para medi-la, além disso é utilizado um aplicativo para dispositivo móvel para controlar e não um assistente de voz.

### **3.5 Skills Alexa relacionadas**

Nesta seção são apresentadas skills para a Alexa já disponíveis na loja da Amazon que estão ligados a automação comercial e automação de forno.

#### **3.5.1 HomeStok**

A skill de Raknahs (c2022) objetiva manter, monitorar e atualizar itens em diferentes categorias como mercearia, medicina, cuidados com o bebê, cuidados pessoais, etc.

O controle de estoque é uma funcionalidade do sistema de automação comercial

que este trabalho visa implementar, porém essa skill, além de ser incompleta para um MEI confeitoiro, ela não está disponível em português.

### **3.5.2 *Group Tasks Manager***

O Group Tasks Manager (GMT) de Sharma (c2022) ajuda a gerenciar uma equipe no trabalho, distribuir tarefas para membros da família ou apenas categorizar listas de tarefas.

Com o GTM é possível:

- Adicionar ou excluir membros únicos da equipe/família;
- Criar listas personalizadas para todos;
- Remover as tarefas depois de concluídas;
- Redefinir todas as listas quando terminar um projeto;

A organização de tarefas pode ser vista como um gerenciamento de agenda, que é uma das funcionalidades do sistema que será implementado neste trabalho, porém esta skill é incompleta para um sistema de automação comercial e não está disponível em português.

### **3.5.3 *Home Connect Oven***

A skill Home Connect Oven (HOME CONNECT GMBH, c2022) permite controlar e monitorar os programas e funções mais importantes do seu forno por meio do Amazon Alexa. Ela possui suporte para pré-aquecimento rápido, termômetro de carne e vários aparelhos e modos de aquecimento. As funcionalidades são ligar o forno diretamente, mudar a temperatura facilmente, mudar a duração do tempo e alternar o modo de aquecimento. Além disso, é possível saber o status de temperatura, a duração e possui um sinal de parada.

A relação dela com este trabalho está na automação de forno, mas ela é compatível apenas com modelos de forno Home Connect da Bosch, Siemens, Neff, Gaggenau. e Thermado e não está disponível em português.

### **3.5.4 *KitchenAid Connected Ovens***

A skill KitchenAid Connected Ovens (WHIRLPOOL CORPORATION, c2022) torna mais fácil interagir, controlar e monitorar fornos da KitchenAid. Possui funcionalidades como pré-aquecer, definir cronômetro e temperatura, além de obter informação do tempo restante no cronômetro do forno.

Também está relacionada com a automação dos equipamentos de cozinha, mas apenas para fornos da marca KitchenAid<sup>2</sup> que têm um custo alto e não está disponível em português.

### 3.6 Relacionando os trabalhos e as *skills* Alexa

A Tabela 1 mostra em que aspectos os trabalhos descritos nas seções 3.1 a 3.5 se relacionam com este.

Tabela 1 – Tabela de comparação dos trabalhos relacionados.

<b>Trabalhos</b>	<b>Automação comercial</b>	<b>Automação de forno</b>	<b>Interface de voz</b>
Protótipo de um sistema de automação para o setor da confeitaria, utilizando a assistente virtual Alexa	Sim	Sim	Sim
Sistema de gestão para microempreendedores	Sim	Não	Não
Sistema de controle de pedidos via web padaria fácil	Sim	Não	Não
Design and implementation of smart cooking based on Amazon Echo	Não	Sim	Sim
Siscont - Sistema de automatização de forno elétrico via dispositivo móvel	Não	Sim	Não

Fonte: elaborada pela autora.

A Tabela 2 mostra a relação das *skills* da Alexa apresentadas na Seção 3.5 com os aspectos abordados neste trabalho.

Tabela 2 – Tabela de *skills* para a Alexa relacionadas.

<b>Skills</b>	<b>Automação comercial</b>	<b>Automação de forno</b>
Protótipo de um sistema de automação para o setor da confeitaria, utilizando a assistente virtual Alexa	Sim	Sim
<i>HomeStok</i>	Sim	Não
<i>Group Tasks Manager</i>	Sim	Não
<i>Home Connect Oven</i>	Não	Sim
KitchenAid Connected Ovens	Não	Sim

Fonte: elaborada pela autora.

<sup>2</sup> <https://www.kitchenaid.com.br/>

## 4 METODOLOGIA

O presente trabalho foi realizado através de pesquisa bibliográfica, de abordagem qualitativa e exploratória. Para alcançar os objetivos do trabalho foram definidas etapas fundamentais para o desenvolvimento de um software: especificação de *software*, projeto e implementação e validação.

A especificação de *software* é a etapa na qual são definidos os serviços requisitados do sistema (SOMMERVILLE, 2011), sendo descrita na Seção 4.1 com levantamento de requisitos. O projeto e implementação do sistema é o processo que converte as especificações do sistema em um sistema executável (SOMMERVILLE, 2011), sendo descrito nas seções 4.1, 4.2 e no capítulo 5. Na seções 4.1 e 4.2 está o projeto do sistema, com modelagem do mesmo contendo a descrição dos casos de uso e os diagramas de requisitos, sequência e de estados. Já no capítulo 5 está o processo de implementação do sistema com o desenvolvimento da *skill*, da arquitetura de comunicação, do servidor *web*, do projeto de banco de dados e do *software* embarcado. A validação de software tem a intenção de mostrar que um software se adequa às especificações do sistema (SOMMERVILLE, 2011), para isto, foi realizado o teste de software descrito na Seção 5.2

### 4.1 Levantamento de requisitos

Os requisitos de um sistema descrevem o que ele deve fazer, os serviços e restrições de seu funcionamento (SOMMERVILLE, 2011). O modelo de levantamento de requisitos descrito por Sommerville separa o processo em quatro etapas: descoberta, classificação e organização, priorização e especificação de requisitos.

Seguindo o modelo de levantamento de requisitos de Sommerville, a primeira etapa a ser realizada é a descoberta de requisitos, no qual são levantadas as necessidades do usuário através das informações reunidas. Para isso, existem técnicas, entre elas a etnografia, uma técnica de observação que pode ser usada para compreender os processos operacionais e ajudar a extrair os requisitos de apoio para esses processos (SOMMERVILLE, 2011). Dessa forma, essa técnica foi aplicada, considerando o conhecimento geral sobre confeitaria e dificuldades de microempreendedores dessa área obtidas pela bibliografia. Além disso a autora utilizou seus conhecimentos na área baseado na observação do trabalho da mãe, que é confeitadeira.

Seguindo às outras etapas, após serem definidos os requisitos, foram classificados em

funcionais e não funcionais, definidas as prioridades entre essenciais e importantes e por último documentados. A Tabela 3 mostra os requisitos levantados, categorizados e com prioridade.

Tabela 3 – Requisitos do sistema.

<b>Requisitos</b>	<b>Categorização</b>	<b>Prioridade</b>
Agendar, pesquisar, excluir e finalizar pedido da agenda	Funcional	Essencial
Consultar agenda do dia	Funcional	Essencial
Adicionar e excluir ingredientes no estoque	Funcional	Essencial
Consultar a quantidade de um ingrediente no estoque	Funcional	Essencial
Definir tempo que produto deve ficar no forno	Funcional	Essencial
Ligar passagem de gás para o forno	Funcional	Essencial
Desligar passagem de gás para o forno	Funcional	Essencial
Ligar luz de dentro do forno quando o produto estiver no forno	Funcional	Importante
Consultar quanto tempo falta até a conclusão	Funcional	Importante
Consultar quais são os produtos que estão no forno	Funcional	Importante
Receber requisições e enviar respostas por interface de voz	Funcional	Essencial
Notificar visualmente quando o tempo definido acabar	Funcional	Essencial
Desligar luz do forno quando o tempo definido acabar	Funcional	Essencial
Desligar passagem de gás para o forno quando o tempo definido acabar	Funcional	Essencial
Consultar a temperatura atual	Funcional	Importante
Ler sensor de temperatura	Funcional	Importante
Se comunicar com o banco de dados para realizar as consultas	Não-funcional	Essencial
Ter uma boa usabilidade	Não-funcional	Essencial
Ter um bom desempenho	Não-funcional	Essencial

Fonte: elaborada pela autora.

## 4.2 Modelagem do sistema

A modelagem do sistema foi dividida em interação com o usuário, com a criação de casos de uso com diálogos entre o usuário e o sistema, e na arquitetura, com os diagramas de requisitos, de sequência e de estados da UML (Linguagem de Modelagem Unificada).

### 4.2.1 Casos de uso

Os casos de uso estão em tabelas divididas em funcionalidades, frase e resposta esperada. O usuário terá uma lista de frases que podem ser ditas para executar a funcionalidade e respostas esperadas para cada uma.

As frases foram criadas para atender às funcionalidades, baseadas no guia de design da Alexa (AMAZON.COM, c2022a), o Alexa Design Guide, seguindo o design de voz para experiências Alexa, o Voice Design for Alexa Experiences, no qual estabelece quatro padrões de

projeto para o projeto situacional:

- Seja adaptável: entender e processar o enunciado adequadamente, em qualquer situação que sua habilidade tenha delineado;
- Seja pessoal: criar familiaridade enquanto fornece informações personalizadas apenas para eles;
- Esteja disponível: criar um design horizontal com prioridade para a voz que mantenha todas as opções abertas para os clientes;
- Seja identificável: a Alexa precisa falar de forma concisa para ajudar o cliente a entender quais informações sua habilidade precisa e para se sentirem confiantes sobre o que está acontecendo;

Essas frases são as *Sample utterances*, enunciados de exemplo que fazem parte do modelo de interação personalizado das habilidades da Alexa no qual é definida a interface de voz. Para definir a interface de voz é necessário mapear a entrada falada para que ela possa ser manipulada no *back-end*. O mapeamento é dividido em:

- Intenções (*Intents*): representa uma ação que atende à solicitação falada de um usuário e pode opcionalmente ter argumentos chamados *slots*;
- Enunciados de exemplo (*Sample utterances*): um conjunto de frases faladas prováveis mapeadas para as intenções;
- Tipos de slots personalizados (*Custom slot types*): uma lista representativa de valores possíveis para um *slot*;

Dessa forma, as funcionalidades aqui apresentadas são as intenções. Dentro das intenções estão os enunciados, no caso as frases, podendo ter os slots, sendo eles variáveis com um tipo definido e representados entre as chaves. As respostas esperadas para cada frase podem conter já uma resposta de execução da funcionalidade ou pode solicitar o valor de algum slot para que possa completar a execução.

#### 4.2.1.1 Consultar agenda

Tabela 4 – Caso de uso consultar agenda.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Consultar agenda	Quero saber minha agenda do dia {data}	- Os pedidos do dia {data} são: [{quantidade} {produto} para {nome do cliente}]. ou - A sua agenda para o dia {data} está vazia
	Agenda de {dia da semana}	
	Quero saber minha agenda de {dia da semana}	
	Qual é a minha agenda do dia {data}	
	Qual é a minha agenda de {dia da semana}	
	Quero saber minha agenda	- Você quer saber a agenda de que dia?

Fonte: elaborada pela autora.

#### 4.2.1.2 Agendar pedidos

Tabela 5 – Caso de uso agendar pedidos.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Consultar agenda	Agendar o pedido de {nome do cliente}	- Para que dia é o pedido? - Qual é o horário de entrega? - Qual é o produto do pedido? - Qual é a quantidade?
	Agendar um pedido	- Tudo bem, qual é o nome do cliente do pedido? - Para que dia é o pedido? - Qual é o horário de entrega? - Qual é o produto do pedido? - Qual é a quantidade?

Fonte: elaborada pela autora.

4.2.1.3 *Pesquisar pedidos do cliente*

Tabela 6 – Caso de uso pesquisar pedidos do cliente.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Consultar agenda	Quais são os pedidos de {nome do cliente}?	- Os pedidos de {nome do cliente} são: [{quantidade} {produto} para {hora} horas] ou - {nome do cliente} não tem pedidos agendados
	Quero saber quais são os pedidos de {nome do cliente}	
	Quais são os pedidos de {nome do cliente} do dia {data}?	- Os pedidos de maria do dia {data} são: [{quantidade} {produto} para {hora} horas]
	Quero saber quais são os pedidos de {nome do cliente} do dia {data}	ou
	Quais são os pedidos de {nome do cliente} de {dia da semana}?	- {nome do cliente} não tem pedidos para o dia {dia}
	Quero saber quais são os pedidos de {nome do cliente} de {dia da semana}	
	Quero saber quais são os pedidos de {nome do cliente} {status do pedido}?	- Os pedidos de {nome do cliente} {status do pedido} são: [{quantidade} {produto} para {hora} horas] ou - {nome do cliente} não tem pedidos {status do pedido}

Fonte: elaborada pela autora.

## 4.2.1.4 Finalizar pedido

Tabela 7 – Caso de uso finalizar pedido.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Finalizar Pedido	Finalize o pedido de {nome do cliente}	<p>- Pedido com {quantidade} {nome do produto} para o dia {data} foi finalizado</p> <p>ou</p> <p>- {nome do cliente} tem {quantidade de pedidos} pedidos que são: pedido {numero do pedido}: {quantidade} {produto} para o dia {data} às {hora} horas]. Qual deseja finalizar?</p> <p>ou</p> <p>{nome do cliente} não tem pedidos agendados</p>
	Quero finalizar um pedido	- De quem é o pedido que deseja finalizar?

Fonte: elaborada pela autora.

#### 4.2.1.5 Excluir pedidos do cliente

Tabela 8 – Caso de uso excluir pedidos do cliente.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Excluir pedidos	Quero excluir o pedido de {nome do cliente}	- Pedido com {quantidade} {nome do produto} para o dia {data} foi excluído ou - {nome do cliente} tem {quantidade de pedidos} pedidos que são: pedido {numero do pedido}: {quantidade} {produto} para o dia {data} às {hora} horas]. Qual deseja excluir?  ou {nome do cliente} não tem pedidos agendados
	Quero excluir um pedido	- De quem é o pedido que deseja excluir?

Fonte: elaborada pela autora.

#### 4.2.1.6 Adicionar ingrediente no estoque

Tabela 9 – Caso de uso adicionar ingrediente no estoque.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Adicionar ingrediente no estoque	Adicione {quantidade} {nome do ingrediente} ao estoque	- Certo. {quantidade} {nome do ingrediente} foi adicionado ao estoque
	Adicione um ingrediente ao estoque	- Tudo bem. Qual é o ingrediente que deseja adicionar? - Certo. E qual é a quantidade?

Fonte: elaborada pela autora.

#### 4.2.1.7 Consultar ingrediente no estoque

Tabela 10 – Caso de uso consultar ingrediente no estoque.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Consultar ingrediente no estoque	Quantos {nome do ingrediente} tem no estoque?	- Tem {quantidade} {nome do ingrediente} no estoque
	Quanto de {nome do ingrediente} tem no estoque?	ou
	Tem {nome do ingrediente} no estoque?	- Não tem {nome do ingrediente} no estoque
	Tem {nome do ingrediente}?	

Fonte: elaborada pela autora.

#### 4.2.1.8 Excluir ingrediente do estoque

Tabela 11 – Caso de uso excluir ingrediente do estoque em caso de consumo ou outro motivo.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Excluir ingrediente do estoque	Exclua {quantidade} {nome do ingrediente} do estoque	- {quantidade} {nome do ingrediente} foi excluído do estoque e restaram {quantidade restante} ou - Não tem {nome do ingrediente} no estoque e restaram {quantidade restante}
	Exclua {nome do ingrediente} do estoque	- Quantos você quer excluir?
	Quero excluir um ingrediente do estoque	- Certo, qual é o ingrediente que você quer excluir?
	Exclua um ingrediente do estoque	- Qual é a quantidade de {ingrediente} que você quer excluir?

Fonte: elaborada pela autora.

#### 4.2.1.9 Ligar passagem de gás para o forno

Tabela 12 – Caso de uso ligar passagem de gás para o forno.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Ligar passagem de gás para o forno	Ligue o forno	- O forno está ligado

Fonte: elaborada pela autora.

#### 4.2.1.10 Desligar passagem de gás para o forno

Tabela 13 – Caso de uso desligar passagem de gás para o forno.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Desligar passagem de gás para o forno	Desligar o forno	- O forno está desligado  ou  - Não foi possível desligar, pois existem produtos no forno

Fonte: elaborada pela autora.

#### 4.2.1.11 Definir tempo que produto deve ficar no forno

Tabela 14 – Caso de uso definir tempo que produto deve ficar no forno.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Definir tempo que produto deve ficar no forno	Marque {tempo} para assar o {nome do produto}	- Pronto. O {nome do produto} foi para o forno às {hora} e deve ficar por {tempo} minutos, finalizando às {hora finalização} horas
	Coloquei o {nome do produto} no forno {hora} e tem que ficar por {tempo}	
	Colocando {nome do produto} no forno	- Quanto tempo deve ficar no forno?
	Coloquei {nome do produto} no forno as {hora}	
	Coloquei {nome do produto} no forno	

Fonte: elaborada pela autora.

#### 4.2.1.12 Consultar quanto tempo falta até a conclusão

Tabela 15 – Caso de uso consultar quanto tempo falta até a conclusão.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Consultar quanto tempo falta até a conclusão	Quanto tempo falta para tirar {nome do produto} do forno	- Faltam {tempo} minutos para tirar o bolo do forno
	Quantos minutos faltam para tirar {nome do produto} do forno	

Fonte: elaborada pela autora.

#### 4.2.1.13 Consultar quais são os produtos que estão no forno

Tabela 16 – Caso de uso consultar quais são os produtos que estão no forno.

<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Consultar a temperatura atual do forno	Quais são os produtos que estão no forno?	- A temperatura atual do forno é {temperatura} graus Celsius. ou - Forno não está ligado

Fonte: elaborada pela autora.

#### 4.2.1.14 Consultar a temperatura atual do forno

Tabela 17 – Caso de uso consultar a temperatura atual do forno.

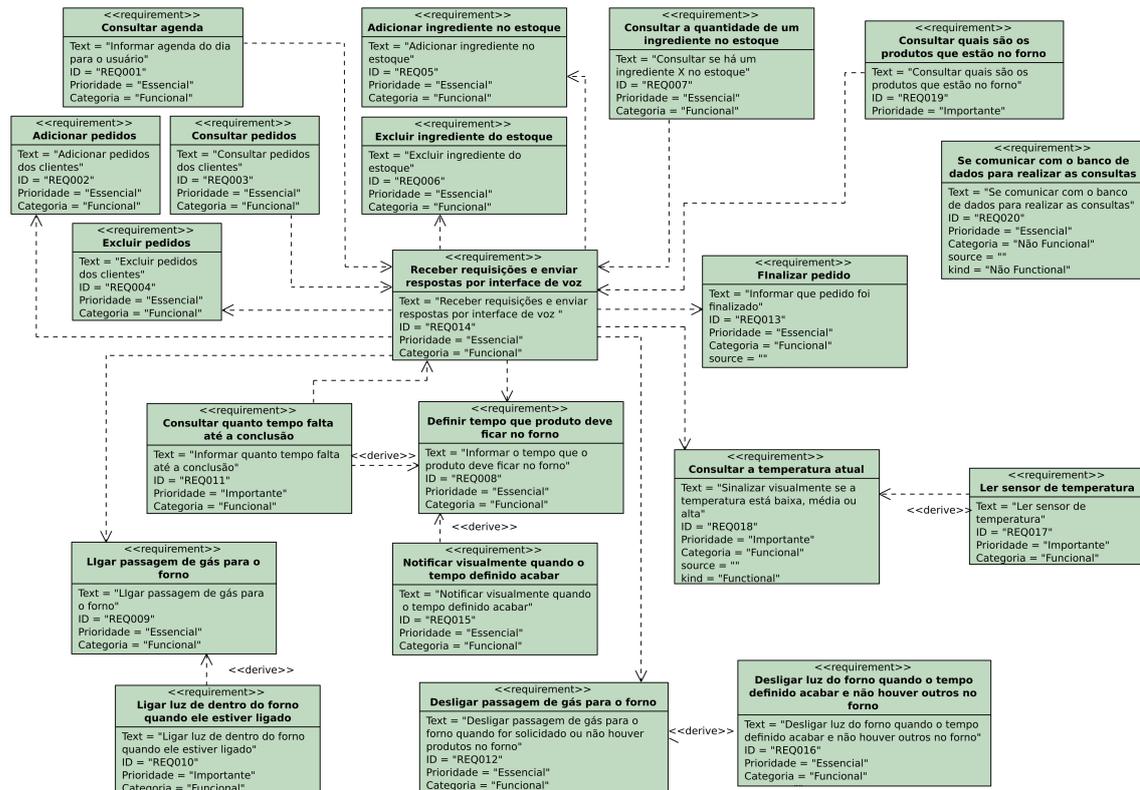
<b>Funcionalidade</b>	<b>Frases</b>	<b>Resposta esperada</b>
Consultar a temperatura atual do forno	Quais são os produtos que estão no forno?	- A temperatura atual do forno é {temperatura} graus Celsius. ou - Forno não está ligado

Fonte: elaborada pela autora.

### 4.2.2 Diagrama de Requisitos

A Figura 3 mostra o diagrama de requisitos da UML com os requisitos levantados na seção 4.1, contendo nome, descrição, categoria, prioridade e correlacionados de acordo com as dependências.

Figura 3 – Diagrama de requisitos.



Fonte: elaborada pela autora.

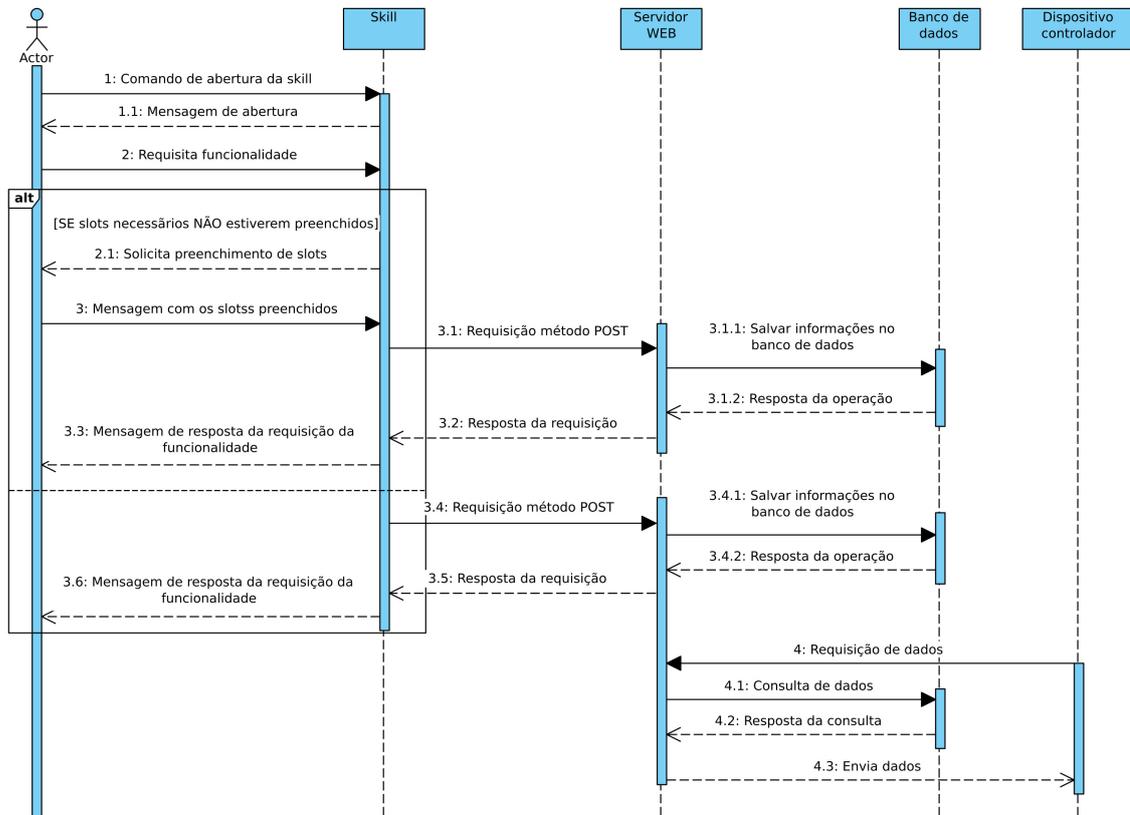
#### 4.2.3 Diagrama de sequência

Para modelar as interações do sistema foi criado um diagrama de sequência, apresentado na Figura 4.

O diagrama foi construído com base do caso de uso que define quanto tempo um produto deve ficar no forno. O modelo contém o ator e os objetos “Skill”, “Servidor Web”, “Banco de Dados” e “Dispositivo controlador”.

Primeiramente o usuário fala a frase de comando de abertura da skill, que devolve uma mensagem de boas vindas através da Alexa. Com a skill aberta, o usuário pode requisitar uma funcionalidade, neste caso, “definir o tempo que produto deve ficar no forno”. Para requisitar essa funcionalidade, como apresentado na Seção 4.2.1.11, o usuário pode pedir para marcar o tempo que o produto deve ficar no forno já passando os valores de todos os slots (tempo e produto), ou pode apenas dizer que colocou o produto no forno e a skill responder pedindo o valor do slot tempo.

Figura 4 – Diagrama de sequência.



Fonte: elaborada pela autora.

Baseado nisso, no diagrama há uma condicional. Se todos os slots necessários não estiverem preenchidos, a skill solicita o preenchimento, pedindo um por vez até que todos tenham um valor. Se todos já estiverem preenchidos, a skill requisita ao servidor web o método de definir o tempo que o produto deve ficar no forno. Esse método faz a comunicação com o banco de dados, realizando a operação de salvar um objeto com o produto, a hora que foi colocado no forno e o tempo que deve ficar. Além disso, atualiza para ligado o valor que indica se o forno está ligado ou não. Após realizar as operações, o servidor responde à skill, que manda uma mensagem por meio da Alexa, confirmando que foi definido um tempo que o produto deve ficar no forno e que horas estará pronto.

Em paralelo a isso, o dispositivo controlador fica fazendo *pooling* a cada 100 ms, se comunicando com o servidor web, que busca no banco de dados o valor que indica se o forno está ligado e se tem notificação. O banco de dados retorna o valor, e assim o servidor devolve o valor que indicará que o forno está ligado e assim acender a luz de dentro dele. Além disso, quando o forno está ligado, é enviado a temperatura para o servidor que envia ao banco de dados.

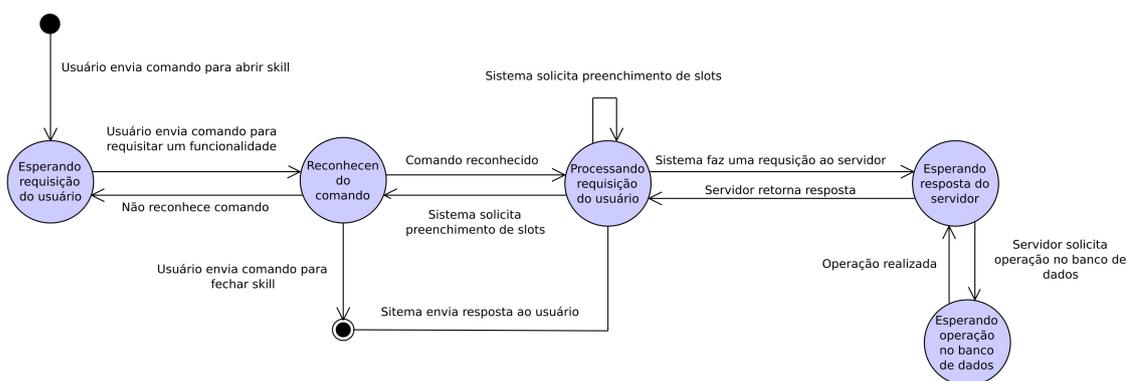
#### 4.2.4 Diagrama de estados

O diagrama de estados mostra os estados de um sistema e os eventos que ativam a transição de um estado para outro (SOMMERVILLE, 2011). A Figura 5 apresenta o diagrama de estados com os estados “Esperando requisição do usuário”, “Reconhecendo comando”, “Processando requisição do usuário”, “Esperando resposta do servidor” e “Esperando resposta do banco de dados”.

O estado “esperando requisição do usuário” é o estado em que a Alexa espera uma requisição para executar alguma funcionalidade da skill. O estado seguinte é o "reconhecendo comando", esse estado em que identifica o que o usuário falou, associando-o a alguma funcionalidade da skill. Após o comando ser reconhecido, vem o estado de “processando a requisição do usuário”, no qual são executadas etapas para tratar o pedido do usuário e devolver uma resposta a ele.

No estado “esperando resposta do servidor” o sistema espera o servidor se comunicar com o banco de dados, realizar a função necessária e devolver uma resposta à skill. O estado “esperando operação no banco de dados” é o que o sistema espera que o banco de dados realize a operação e depois devolva a resposta da operação.

Figura 5 – Diagrama de estados.



Fonte: elaborada pela autora.

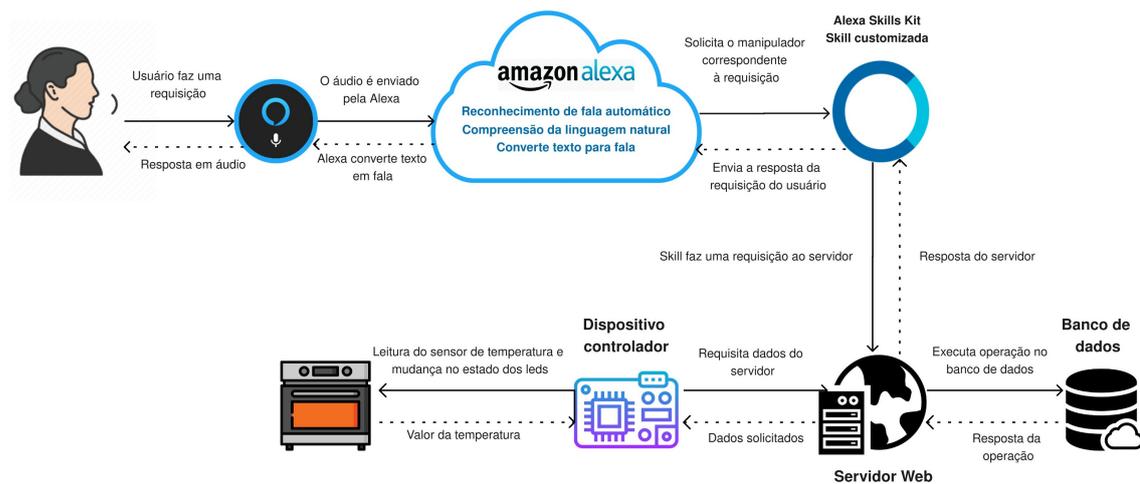
## 5 DESENVOLVIMENTO

### 5.1 Desenvolvimento do sistema

Após o levantamento de requisitos e modelagem, a próxima etapa foi a implementação do sistema. O primeiro estágio do processo de implementação foi o desenvolvimento do modelo de arquitetura, responsável por identificar os principais componentes estruturais e os relacionamentos entre eles e descrever como o sistema está organizado.

Desse modo, foi desenvolvido o modelo de arquitetura como mostra a Figura 6. O usuário se comunica por voz através um dispositivo Alexa, requisitando uma funcionalidade. O dispositivo Alexa por sua vez envia o fluxo de áudio recebido para ser reconhecido pelo serviço Alexa na nuvem. Depois de reconhecido, é solicitado o manipulador correspondente à requisição. Este manipulador é o código da *skill* que executa a ação solicitada e retorna uma resposta apropriada. Para executar a ação, é realizada uma requisição ao servidor web, que se comunica com o banco de dados e executa uma operação determinada. O servidor então fornece uma resposta à *skill* para que ela responda à requisição do usuário. Como dito anteriormente na Seção 4.2.3, o dispositivo controlador fica fazendo o *pooling*, requisitando dados e enviando o valor da temperatura ao servidor.

Figura 6 – Modelo de arquitetura do sistema.



Fonte: elaborada pela autora.

### 5.1.1 Dispositivo controlador

O dispositivo controlador é a parte do sistema que faz a automação do forno. Os requisitos desta parte são ligar e desligar o forno através da liberação e bloqueio da passagem de gás, medir temperatura do forno, ligar luz enquanto houver produto no forno e notificar visualmente quando o tempo no forno de algum produto acabar.

Para construir o protótipo foram selecionados os componentes necessários, começando pelo microcontrolador ESP-32 <sup>1</sup>, escolhido por ter módulo de wi-fi integrado e pela facilidade de programação através da IDE do Arduino, além do baixo consumo energético.

A liberação e o bloqueio da passagem do gás para o forno pode ser feita utilizando uma válvula solenóide, um dispositivo eletromecânico utilizado para o controle de fluxo de líquidos e gases, possibilitando a abertura e o fechamento da tubulação por onde o gás passa. Para fazer essa liberação e bloqueio, é necessário um relé, que atua como uma chave, possuindo o estado ligado e desligado.

Como o relé possui dois estados, assim como um *led* que pode ser aceso e apagado, foi decidido usar dois *leds* no protótipo para representar os dois estados do relé, com o objetivo de facilitar o desenvolvimento e testes. O *led* verde indica que está ligado e o vermelho indica que está desligado. Posteriormente, os *leds* podem ser substituídos pelo relé e o solenóide, sem alterar a lógica da ação.

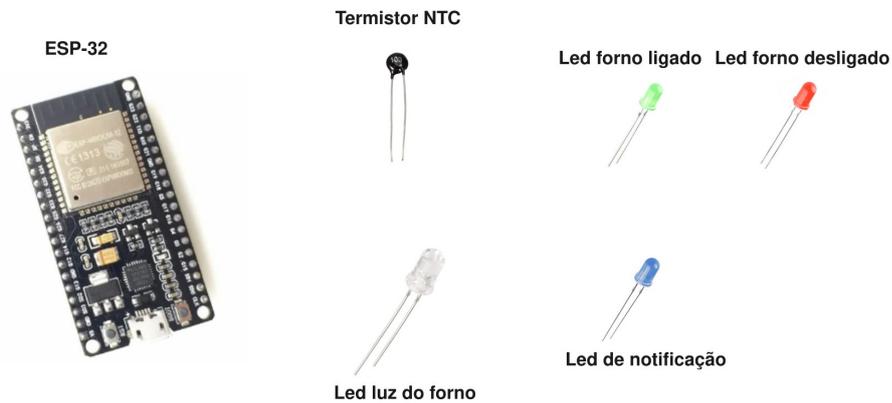
Para medir a temperatura foi decidido usar um termistor NTC de 10k $\Omega$ , um tipo de resistor cujo valor varia de acordo com a temperatura. Um termistor do tipo NTC (*Negative Temperature Coefficient*), quer dizer que a sua resistência diminui conforme a temperatura aumenta. A sua faixa de medição é de -55 a 125°C e mesmo que a temperatura máxima de um forno seja superior, ele foi escolhido apenas para a prova de conceito do protótipo, por ser um componente de baixo custo e fácil de ser encontrado. Posteriormente ele pode ser substituído por um sensor que suporte temperaturas mais altas sem precisar alterar a lógica.

Por fim, para representar a luz que liga quando houver produto no forno e notificar visualmente que o tempo de algum produto acabou, foi decidido usar *leds* branco e azul respectivamente. Todos os componentes utilizados estão ilustrados na Figura 7.

---

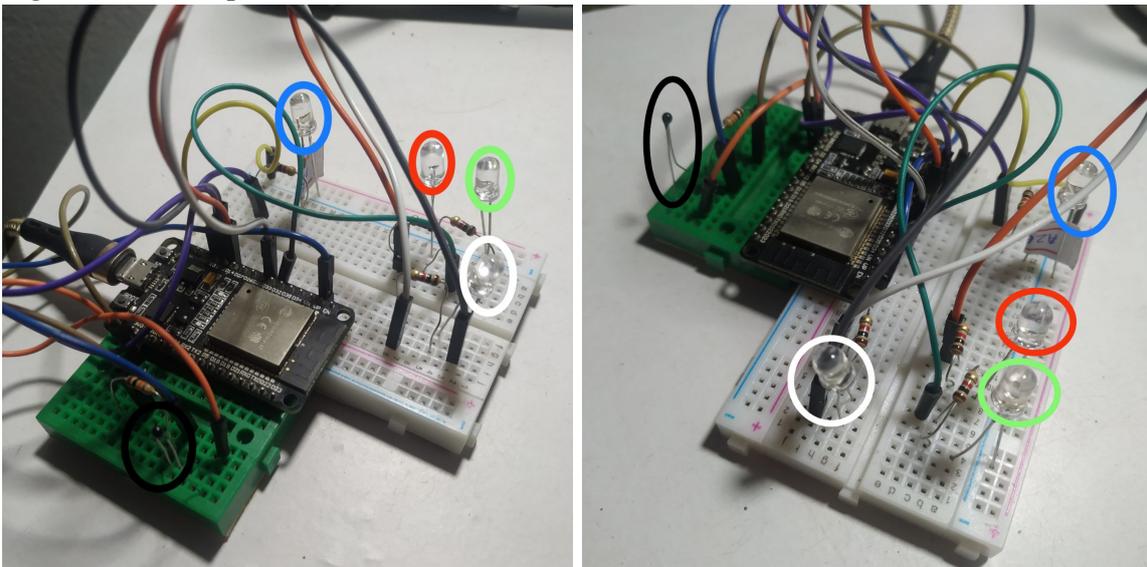
<sup>1</sup> <https://www.espressif.com/en/products/socs/esp32>

Figura 7 – Componentes do sistema.



Fonte: elaborada pela autora.

Figura 8 – Protótipo.



Fonte: elaborada pela autora.

A Figura 8 mostra o protótipo montado, nela estão circulos os leds, com a cor correspondente e o termistor circulado em preto para melhor identificação.

### 5.1.2 Skill

“As skills da Alexa são aplicativos ativados por voz que adicionam recursos ao dispositivo compatível com a Alexa” (AMAZON.COM, c2022b). A primeira etapa é o desenvolvimento do modelo de interação, descrito na Seção 4.2.1. Além disso, é necessário criar um nome de invocação da skill, para que os usuários usem para abri-la. O nome escolhido foi “minha confeitaria”, logo para abrir a skill, o usuário precisa dizer “abra minha confeitaria”.

Como definido nesta Seção de casos de uso, as funcionalidades lá apresentadas são as intenções, logo, a *skill* tem as seguintes intenções:

- **ConsultarAgendaIntent:** intenção de quando o usuário quer saber a sua agenda de uma determinada data. Seus enunciados são as frases descritas na Seção 4.2.1.1. Possui apenas um *slot* de nome “data”, tipo *AMAZON.DATE* e preenchimento obrigatório. Por ser do tipo *AMAZON.DATE*, a data pode ser preenchida com o dia da semana ou com uma data específica como "hoje", "agora" ou "vinte e cinco de novembro";
- **AgendarPedidoIntent:** intenção de quando o usuário quer agendar um pedido de um determinado cliente. Seus enunciados são as frases descritas na Seção 4.2.1.2. Possui os *slot* “nomeCli”, do tipo *AMAZON.FirstName*, “data”, do tipo *AMAZON.DATE*, “hora”, do tipo *AMAZON.TIME*, “produto”, do tipo *AMAZON.Food* e “quantidade”, do tipo *AMAZON.NUMBER*, sendo todos eles de preenchimento obrigatório;
- **PesquisarPedidosClienteIntent:** intenção de quando o usuário quer saber os pedidos de um determinado cliente. Seus enunciados são as frases descritas na Seção 4.2.1.3. Possui os *slots* “nomeCli”, do tipo *AMAZON.FirstName*, “data”, do tipo *AMAZON.DATE*, e “statusPedido”, do tipo *StatusPedido*, um *slot* personalizado com os valores “AGENDADO” e “FINALIZADO”. Apenas “nomeCli” é de preenchimento obrigatório;
- **FinalizarPedidoIntent:** intenção de quando o usuário quer finalizar o pedido de um determinado cliente. Seus enunciados são as frases descritas na Seção 4.2.1.4. Possui o *slot* “nomeCli” do tipo *AMAZON.FirstName* e preenchimento obrigatório;
- **ExcluirPedidoIntent:** intenção de quando o usuário quer excluir o pedido de um determinado cliente. Seus enunciados são as frases descritas na Seção 4.2.1.5. Possui o *slot* “nomeCli” do tipo *AMAZON.FirstName* e preenchimento obrigatório;
- **AdicionarIngredienteEstoqueIntent:** intenção de quando o usuário quer adicionar um ingrediente no estoque. Seus enunciados são as frases descritas na Seção 4.2.1.6. Possui os *slots* “quantidade”, do tipo *AMAZON.NUMBER* e “ingrediente”, do tipo *AMAZON.Food*, os dois com preenchimento obrigatório;

- **ConsultarIngredienteEstoqueIntent:** intenção de quando o usuário quer saber se tem um ingrediente no estoque ou a quantidade. Seus enunciados são as frases descritas na Seção 4.2.1.7. Possui o *slot* “ingrediente”, do tipo *AMAZON.Food* e preenchimento obrigatório;
- **ExcluirIngredienteEstoqueIntent:** intenção de quando o usuário quer excluir ingredientes do estoque por que ele foi usado ou por qualquer outro motivo. Seus enunciados são as frases descritas na Seção 4.2.1.8. Possui os *slots* “quantidade”, do tipo *AMAZON.NUMBER* e preenchimento obrigatório e “ingrediente”, do tipo *AMAZON.Food*;
- **LigarFornoIntent:** intenção de quando o usuário quer ligar a passagem de gás para o forno. Seus enunciados são as frases descritas na Seção 4.2.1.9 e não possui *slots*.
- **DesligarFornoIntent:** intenção de quando o usuário quer desligar a passagem de gás para o forno. Seus enunciados são as frases descritas na Seção 4.2.1.10 e não possui *slots*;
- **DefinirTempoFornoIntent:** intenção de quando o usuário quer definir o tempo que um determinado produto deve ficar no forno. Seus enunciados são as frases descritas na Seção 4.2.1.11. Possui os *slots* “hora”, do tipo *AMAZON.TIME*, “tempo” do tipo *AMAZON.Duration* e preenchimento obrigatório e “produto”, do tipo *AMAZON.Food*.
- **ConsultarTempoRestanteIntent:** intenção de quando o usuário deseja saber quanto tempo resta para que um determinado produto termine de assar e possa ser retirado do forno. Seus enunciados são as frases descritas na Seção 4.2.1.12. Possui apenas o *slot* “produto”, do tipo *AMAZON.Food*.
- **ConsultarProdutosFornoIntent:** intenção de quando o usuário deseja saber quais são os produtos que estão no. Seus enunciados são as frases descritas na Seção 4.2.1.13 3 não possui *slots*.
- **ConsultarTemperaturaIntent:** intenção de quando o usuário quer saber qual é a temperatura atual do forno. Seus enunciados são as frases descritas na Seção 4.2.1.14 e não possui *slots*.
- **EmergenciaIntent:** intenção de quando o usuário fala que houve uma emergência e assim, o forno é desligado.

- **AjudaIntent:** intenção de quando o usuário pede ajuda para saber mais funcionalidades e como solicitá-las.

Além dessas intents, tem as padrões da Alexa, a *AMAZON.CancelIntent*, no qual o usuário pode cancelar uma transação ou sair da *skill*, a *AMAZON.RepeatIntent* para quando o usuário quiser que a Alexa repita o que falou anteriormente. Além disso, a *AMAZON.FallbackIntent*, acionada quando o que o usuário falou não corresponde a nenhuma das intenções existentes.

Cada intenção possui um manipulador, uma função com a lógica para executar a ação solicitada por ela. Desse modo, a *skill* possui as funções:

- **ConsultarAgendaIntentHandler:** esta função recebe o valor do *slot* “data” e passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, buscando os pedidos da data solicitada e os envia como resposta. Quando recebe a resposta do servidor, a função trata os dados recebidos, formulando um texto de saída com os pedidos da data ou com uma mensagem de que não há pedidos, como descrito em resposta esperada na Seção 4.2.1.1;
- **AgendarPedidoIntentHandler:** esta função recebe os valores dos *slots* “nomeCli”, “data”, “hora”, “produto” e “quantidade” e os passa como parâmetro para a requisição ao servidor web, juntamente com o valor de status "AGENDADO". O servidor então acessa o banco de dados, salvando esses valores como atributos de um pedido e envia uma resposta com um código de aprovação. Quando recebe a resposta do servidor, a função formula um texto de saída confirmando que o pedido foi adicionado, como descrito em resposta esperada na Seção 4.2.1.2;
- **PesquisarPedidosClienteIntentHandler:** esta função recebe o valor dos *slot* “nomeCli”, “data” e “statusPedido” e passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, buscando os pedidos do cliente, data e status solicitados e os envia como resposta, se o statusPedido não houver valor, a busca será dos pedidos com o status “AGENDADO”. Quando recebe a resposta do servidor, a função trata os dados recebidos, formulando um texto de saída com os pedidos da data ou com uma mensagem de que não há pedidos, como descrito em resposta esperada na Seção 4.2.1.3;
- **FinalizarPedidoIntentHandler:** esta função recebe o valor dos *slot* “nomeCli” e passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, buscando os pedidos do cliente solicitado com status

- de agendado e os envia como resposta. Quando recebe a resposta do servidor, a função verifica se o cliente tem mais de um pedido agendado, se não tiver, a função faz outra requisição ao servidor para mudar o status do pedido para “FINALIZADO” e responde ao usuário com a mensagem de pedido finalizado. Se tiver mais de um pedido, a resposta de saída lista os pedidos do cliente e pergunta qual o usuário quer finalizar. Quando o usuário escolhe o pedido, é feita outra solicitação ao servidor para mudar o status dele para “FINALIZADO” e responde ao usuário a mensagem de que o pedido foi finalizado;
- **ExcluirPedidoIntentHandler:** esta função recebe o valor dos *slot* “nomeCli” e passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, buscando os pedidos do cliente solicitado com status de agendado e os envia como resposta. Quando recebe a resposta do servidor, a função verifica se o cliente tem mais de um pedido agendado, se não tiver, a função faz outra requisição ao servidor para excluir o pedido e responde ao usuário com a mensagem de pedido excluído. Se tiver mais de um pedido, a resposta de saída lista os pedidos do cliente e pergunta qual o usuário quer excluir. Quando o usuário escolhe o pedido, é feita outra solicitação ao servidor para excluí-lo e responde ao usuário a mensagem de que o pedido foi excluído;
  - **AdicionarIngredienteEstoqueIntentHandler:** esta função recebe os valores dos *slots* “quantidade” e “produto” e os passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, salvando esses valores como atributos de um ingrediente no estoque e envia uma resposta com um código de aprovação. Quando recebe a resposta do servidor, a função formula um texto de saída confirmando que o ingrediente foi adicionado, como descrito em resposta esperada na Seção 4.2.1.6;
  - **ConsultarIngredienteEstoqueIntentHandler:** esta função recebe o valor do *slot* “ingrediente” e passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, buscando a quantidade do ingrediente solicitado e envia como resposta. Quando recebe a resposta do servidor, a função trata os dados recebidos, formulando um texto de saída com a quantidade do ingrediente, como descrito em resposta esperada na Seção 4.2.1.7;
  - **ExcluirIngredienteEstoqueIntentHandler:** esta função recebe os valores dos

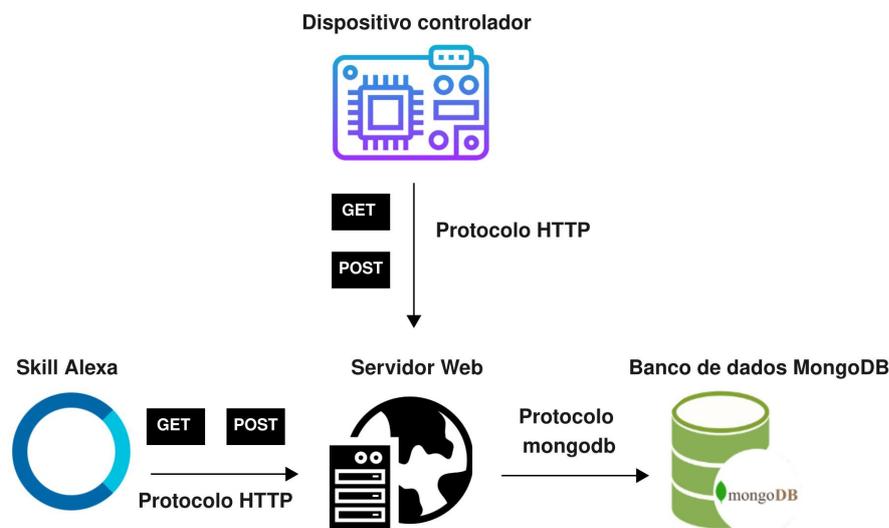
- slots* “quantidade” e “ingrediente” e os passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, buscando a quantidade do ingrediente solicitado e subtrai dela o valor do *slot* “quantidade” e envia uma resposta com um código de aprovação. Quando recebe a resposta do servidor, a função trata os dados recebidos, formulando um texto de saída com a mensagem de que foi excluído, como descrito em resposta esperada na Seção 4.2.1.8;
- **LigarFornoIntentHandler:** esta função faz uma requisição ao servidor web, que acessa o banco de dados, atualizando o status do forno para 1 e responde com um código de aprovação. A função então responde ao usuário uma mensagem de que o forno foi ligado, como descrito em resposta esperada na Seção 4.2.1.9;
  - **DesligarFornoIntentHandler:** esta função faz uma requisição ao servidor web, que acessa o banco de dados, atualizando o status do forno para 0 e responde com um código de aprovação caso não haja produtos no forno. A função então responde ao usuário uma mensagem de que o forno foi desligado ou que não foi possível desligar, como descrito em resposta esperada na Seção 4.2.1.10;
  - **DefinirTempoFornoIntentHandler:** esta função recebe os valores dos *slots* “produto”, “tempo” e “hora”, se “hora” estiver vazio, recebe a hora atual. Depois disso, esses valores são passados como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, salva esses valores como atributos de um objeto de produto no forno, muda o status do forno para ligado e envia uma resposta com um código de aprovação. Quando recebe a resposta do servidor, a função calcula a hora de finalização e formula um texto de saída com o tempo que vai ficar e a hora de finalização, como descrito em resposta esperada na Seção 4.2.1.11;
  - **ConsultarTempoRestanteIntentHandler:** esta função recebe o valor do *slot* “produto” e passa como parâmetro para a requisição ao servidor web. O servidor então acessa o banco de dados, buscando o tempo que o produto solicitado deve ficar no forno e envia como resposta. Quando recebe a resposta do servidor, a função calcula o tempo restante e formula um texto de saída com ele, como descrito em resposta esperada na Seção 4.2.1.12;
  - **ConsultarTempoRestanteIntentHandler:** esta função faz uma requisição ao servidor web, que acessa o banco de dados, buscando os produtos que estão no

- forno e envia como resposta. A função então responde ao usuário uma mensagem com a temperatura, como descrito em resposta esperada na Seção 4.2.1.14;
- **ConsultarTemperaturaIntent:** esta função faz uma requisição ao servidor web, que acessa o banco de dados, buscando o valor da temperatura do forno e envia como resposta. A função então responde ao usuário uma mensagem com a temperatura, como descrito em resposta esperada na Seção 4.2.1.14;
  - **EmergenciaIntentHandler:** esta função faz uma requisição ao servidor web, que acessa o banco de dados, atualizando o status do forno para 0 e envia a resposta de que o forno foi desligado.
  - **AjudaIntentHandler:** esta função retorna ao usuário uma mensagem com as funcionalidades do sistema e pergunta sobre qual o usuário quer saber mais, o usuário então responde qual é a funcionalidade, ela é identificada e uma mensagem é enviada com as instruções de uso.

### 5.1.3 Arquitetura da comunicação

Para o funcionamento do sistema, faz-se necessário uma comunicação entre os componentes. A arquitetura de comunicação escolhida foi o cliente-servidor, no qual é baseada no protocolo requisição-resposta, o cliente requisita ao servidor e ele retorna uma resposta ao cliente.

Figura 9 – Arquitetura da comunicação.



Fonte: elaborada pela autora.

A Figura 9 ilustra a arquitetura do sistema, as requisições ao servidor web são realizadas através do protocolo HTTP, utilizando os métodos *GET* ou *POST*. O banco de dados escolhido para o projeto foi o *mongoDB*, portanto, a comunicação com ele é realizada através do protocolo *mongodb*.

#### 5.1.4 Servidor Web

Para a criação do servidor web, foi utilizado o Node.js com o Express.js. O Node.js é o ambiente de execução Javascript que permite que a aplicação do servidor execute na máquina. Para tornar a aplicação em um servidor web, foi utilizado o Express.js, um *framework* do node que fornece os recursos necessários, como o gerenciamento de requisições http.

O primeiro passo na criação do servidor é fazê-lo escutar as requisições. Para isso, foi utilizado o método *listen()* do *express* e definido uma porta para o que servidor possa receber as requisições do cliente. Para que os clientes possam enviar as requisições, o servidor precisa estar executando em uma máquina que possa ser acessada remotamente. Neste projeto foi utilizada a plataforma Heroku<sup>2</sup>, na qual é possível hospedar o servidor para que ele seja acessado através do protocolo HTTP, como definido na arquitetura de comunicação.

Como o protocolo é o HTTP, o servidor utiliza os métodos *get()* e *post()* fornecidos pelo *framework express*. Esses métodos são chamados de métodos de roteamento, tendo como parâmetro um caminho de rota, que combinado com o método de requisição, define terminais para que as requisições sejam atendidas. Desse modo, o servidor é composto por funções de retorno de chamada, que possuem um método de roteamento e os objetos de requisição e resposta como argumento. Este servidor web atua como uma API (*Application Programming Interface*) se comunicando com o banco de dados, então as ações de suas funções são realizar uma operação no banco de dados e retornar a resposta. Nesse caso, a operação realizada no banco é a de salvar um pedido.

#### 5.1.5 Banco de Dados

O banco de dados escolhido foi o *MongoDB*<sup>3</sup>, um banco de dados *open source* e *NoSql*, orientado a documentos. Por ser *NoSql*, ao invés de tabelas, ele possui coleções, documentos e campos. As coleções são como as tabelas, que possuem documentos e os documentos

---

<sup>2</sup> <https://www.heroku.com>

<sup>3</sup> <https://www.mongodb.com/>

são conjuntos de pares de valores-chave que são os campos.

No nosso banco de dados temos então as coleções: “pedidos”, “estoque”, “produtoNoForno” e “forno”. Os documentos de pedidos possuem os campos “nomeCli”, “data”, “hora”, “produto” e “quantidade”, os de estoque possuem “quantidade” e “ingrediente” e os de *produtoNoForno* possuem “nomePro”, “hora” e “tempo”.

Já a coleção forno, tem apenas um documento e armazena as informações do forno. Seus campos são “temperatura”, “statusForno” e “statusNotificação”, sendo status com valor 0 ou 1, referente ao estado ligado ou desligado e notificação também com valor 0 ou 1, para saber se precisa notificar que acabou o tempo de um produto no forno.

Para que o servidor se comunique com banco de dados, foi criado o banco de dados em nuvem com o *MongoDB Atlas*, o banco de dados em nuvem do mongoDB. Dessa forma, o servidor se conecta ao banco utilizando o protocolo *mongodb* e realiza as operações nele.

### 5.1.6 Software embarcado

O desenvolvimento do software para o microcontrolador ESP-32 foi realizado usando o Arduino IDE <sup>4</sup>(*Arduino Integrated Development Environment*), um ambiente de desenvolvimento que facilita o desenvolvimento e a gravação de códigos diretamente no microcontrolador.

O software deve ser capaz de realizar as funções de ligar ou desligar os *leds* que representam o status do forno, a luz do forno e o de notificação, além de ler o sensor de temperatura. Para realizar essas funções, é necessário obter as informações que estão armazenadas no banco de dados, na coleção forno, como relatado na Seção 5.1.5.

Para obter essas informações, é necessário fazer uma requisição ao servidor web para que ele busque essas informações no banco de dados. Desse modo, o primeiro passo na construção do software embarcado é estabelecer uma conexão com o servidor web. Primeiro a ESP precisa estar conectada a internet, então foi utilizado o método *WIFI.begin()* da biblioteca “wifi.h”, com o nome da rede e a senha como parâmetros. Se a conexão com a internet for estabelecida, o método *getInfosForno()* é chamado.

No método *getInfosForno()*, é instanciado a variável http da classe HTTPCliente, para trabalhar com as funcionalidades do HTTP. A requisição ao servidor é feita utilizando o método *http.begin()*, passando a URL (*Uniform Resource Locator*) do servidor como parâmetro e o *http.GET()* para enviar a requisição. A função do servidor dessa requisição é a

<sup>4</sup> <https://www.arduino.cc/en/software>

“app.get('/getInfosForno', (req, res) => )”, além de buscar os dados da coleção forno, ela também checa quanto tempo resta para retirar um produto do forno, se acabar o tempo de algum, o valor de *statusNotificacao* passa a ser 1.

O resultado de *http.GET()* é armazenado em uma variável, no qual será verificado o seu valor, se for maior que 0, a requisição foi bem sucedida e a resposta precisa ser desserializada para obter os valores de “statusForno” e “statusNotificacao”. Após isso são chamadas às funções *ligaEDesligaForno()*, *notificar()* e *enviarValorTemperatura()*.

A função *ligaEDesligaForno()* verifica o valor de “statusForno”, se for 1, acende o *led* verde e o da luz do forno e apaga o vermelho, se for 0, acende o *led* vermelho e apaga o verde e o da luz do forno.

Em “notificar()”, o valor de “statusNotificacao” é verificado, se for 1, acende o *led* e deixa ele ligado por um minuto e faz uma requisição ao servidor, enviando o valor 0 para notificação, fazendo uma solicitação à função “app.get('/setInfosForno', (req, res) => )”, que salva no banco de dados o novo valor de “statusNotificacao”. Se for 0, o *led* é apagado.

Na função *enviarValorTemperatura()*, o valor da temperatura atual é obtido através da leitura do termistor, com uma biblioteca chamada “termistor.h”. Se o forno estiver ligado, o valor da temperatura é enviado através da função “app.get('/setInfosForno', (req, res) => )”, que salva o valor da temperatura no banco de dados.

## 5.2 Teste de software

“Teste de software é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado.” (NETO; CLAUDIO, 2007). O objetivo é detectar falhas para que elas sejam identificadas e corrigidas.

Para testar o sistema foi utilizado o teste funcional, uma técnica no qual os casos de teste são desenhados para garantir que os requisitos e especificações do sistema tenham sido atendidos projetados. Nela o sistema é considerado uma caixa-preta, isto quer dizer que são fornecidos dados de entrada, os testes executados e o resultado obtido é comparado ao resultado esperado, se forem iguais, o teste foi bem sucedido.

Nesta técnica não são considerados os detalhes de implementação e o software é avaliado do ponto de vista do usuário (DELAMARO, 2016). Desse modo, os casos de testes foram criados baseados nos casos de uso, visto que eles descrevem os requisitos, representam

como o sistema será utilizado e permite identificar os caminhos que o usuário pode percorrer para conseguir o que deseja.

Um caso de teste tem o objetivo de exercitar um caminho particular de um programa ou verificar sua conformidade com um requisito específico (HEUMANN, 2001). O processo para criá-los pode ser descrito em três etapas: para cada caso de uso deve ser gerado um conjunto completo de cenários, em cada cenário deve ser identificado pelo menos um caso de teste e suas condições e por fim identificar valores de dados para testar.

Os cenários são os fluxos dos casos de uso, para cada caso de uso apresentado na Seção 4.2.2 foi criada uma tabela como a Tabela 18, que estão no Apêndice A e realizado os testes utilizando valores para cada variável descrita entre chaves.

Tabela 18 – Caso de teste para o caso de uso consultar agenda.

Caso de uso: Consultar agenda	
Fluxo básico - <i>Slot</i> {data} está preenchido e existem pedidos para o dia	<ol style="list-style-type: none"> <li>1. Usuário pede a agenda do dia <ul style="list-style-type: none"> <li>- Quero saber minha agenda do dia {data}</li> </ul> </li> <li>2. Alexa responde a agenda do dia: <ul style="list-style-type: none"> <li>- Os pedidos do dia {data} são: <ul style="list-style-type: none"> <li>[[quantidade] {produto} para {nome do cliente}]]</li> </ul> </li> </ul> </li> </ol>
Fluxo alternativo 1 - <i>Slot</i> {data} não está preenchido e existem pedidos para o dia	<ol style="list-style-type: none"> <li>1. Usuário pede a agenda do dia <ul style="list-style-type: none"> <li>- Quero saber minha agenda</li> </ul> </li> <li>2. Alexa pergunta de que dia o usuário quer saber a agenda <ul style="list-style-type: none"> <li>- A sua agenda para o dia {data} está vazia</li> </ul> </li> <li>3. Usuário responde o dia <ul style="list-style-type: none"> <li>- {dia}</li> </ul> </li> <li>4. Alexa responde a agenda do dia <ul style="list-style-type: none"> <li>- Os pedidos do dia {data} são: <ul style="list-style-type: none"> <li>[[quantidade] {produto} para {nome do cliente}]]</li> </ul> </li> </ul> </li> </ol>
Fluxo alternativo 2 - <i>Slot</i> {data} está preenchido e NÃO existem pedidos para o dia	<ol style="list-style-type: none"> <li>1. Usuário pede a agenda do dia: <ul style="list-style-type: none"> <li>- Quero saber minha agenda do dia {data}</li> </ul> </li> <li>2. Alexa responde que a agenda do dia está vazia: <ul style="list-style-type: none"> <li>- A sua agenda para o dia {data} está vazia.</li> </ul> </li> </ol>
Fluxo alternativo 3 - <i>Slot</i> {data} NÃO está preenchido e NÃO existem pedidos para o dia	<ol style="list-style-type: none"> <li>1. Usuário pede a agenda do dia <ul style="list-style-type: none"> <li>- Quero saber minha agenda</li> </ul> </li> <li>2. Alexa pergunta de que dia o usuário quer saber a agenda <ul style="list-style-type: none"> <li>- A sua agenda para o dia {data} está vazia</li> </ul> </li> <li>3. Usuário responde o dia <ul style="list-style-type: none"> <li>- {dia}</li> </ul> </li> <li>4. Alexa responde que a agenda do dia está vazia: <ul style="list-style-type: none"> <li>- A sua agenda para o dia {data} está vazia.</li> </ul> </li> </ol>

Fonte: elaborada pela autora.

Além de verificar se os requisitos e as funcionalidades do sistema foram atendidas, estes testes foram usados para calcular a latência do sistema. Cada fluxo representa uma requisição à Alexa, que por sua vez, leva um tempo para ser realizada e devolver a resposta ao usuário, portanto, esse tempo precisará ter um valor no qual o usuário não precise esperar muito tempo pela resposta.

Com isso, para cada cada fluxo básico dos casos de uso, foi calculado o tempo de resposta e coletada uma amostra com 10 tempos. Depois disso, o cálculo final da latência foi obtido a partir da média desses 10 tempos.

### 5.2.1 Resultados e análise dos testes

Após a realização de todos os testes, foi obtido o resultado de que todos os casos de teste passaram, ou seja, todos os fluxos deles obtiveram a resposta esperada, exceto o teste para o caso de uso de pesquisar pedidos do cliente e de adicionar um ingrediente no estoque.

No cálculo da latência foram obtidos os resultados apresentados na Tabela 19.

Tabela 19 – Tempos de latência de cada caso de uso.

<b>Caso de uso</b>	<b>Tempo</b>
Consultar agenda	81.1 ms
Agendar pedidos	87.4 ms
Pesquisar pedidos do cliente	80.8 ms
Finalizar pedido	162.2 ms
Excluir pedidos do cliente	161.8 ms
Adicionar ingrediente no estoque	81.2 ms
Consultar ingrediente no estoque	79.8 ms
Excluir ingrediente do estoque	148.8 ms
Ligar passagem de gás para o forno	80.2 ms
Desligar passagem de gás para o forno	144 ms
Definir tempo que produto deve ficar no forno	88.6 ms
Consultar quanto tempo falta até a conclusão	82.4 ms
Consultar quais são os produtos que estão no forno	77.5 ms
Consultar a temperatura atual do forno	144.7 ms

Fonte: elaborada pela autora.

Em alguns testes iniciais, foram obtidos resultados em torno de 200 a 300 ms, mas os resultados seguintes mostraram uma latência em torno de 80 milissegundos e alguns são superiores a 100 milissegundos como apresentado na Tabela 19.

Os casos em que o valor é maior que 100 ms, deve-se ao fato de que para conseguir a resposta adequada são realizadas mais de uma operação no banco de dados. Foi calculado

também o tempo que leva para o software embarcado obter os dados e realizar suas funções, obtendo resultados em torno de 1.5 s.

De acordo com Jakob Nielsen (1993), o conselho básico sobre os tempos de resposta para softwares de computadores têm sido o mesmo nos últimos 30 anos:

- 100 ms é o limite para que o usuário sinta que o sistema está respondendo instantaneamente;
- 1 s é aproximadamente o limite para que fluxo de pensamento do usuário permaneça ininterrupto, mesmo que o usuário perceba o atraso;
- 10 s é o limite para manter a atenção do usuário focada no diálogo;

Em 2014, Jakob atualizou que as diretrizes para aplicativos web são as mesmas de todos os aplicativos, estabelecidas há 46 anos. Mesmo que estes tempos não tenham sido estabelecidos para aplicações de voz, podem ser levados em consideração já que são baseados nas percepções do usuário.

Dessa forma, pode-se concluir que como o tempo máximo da latência foi 1.5 s e a maioria dos tempos foi abaixo de 100 s, o sistema possui um tempo aceitável, de forma que o usuário sinta que o sistema está respondendo instantaneamente mesmo que em algum momento possa perceber um atraso. Com os testes apresentando resultados positivos, é possível também dizer que o sistema funcionou corretamente e atendeu aos requisitos especificados.

Portanto, como o sistema atendeu aos requisitos, é possível dizer que ele foi validado, visto que a validação de um software mostra que ele se adequa às especificações do sistema e a principal técnica é executar testes com dados simulados.

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho foi proposto pensando nos microempreendedores individuais, levando em conta sua importância na economia brasileira, suas dificuldades e conhecimento geral sobre suas atividades, onde foi desenvolvido um sistema de automação que gerencia agenda, estoque e automação do forno. Para tal foi criado uma *skill*, permitindo usar o sistema utilizando a voz o que proporciona praticidade, economia de tempo já que não será preciso pegar no aparelho celular para dar os comandos, maior higiene visto que reduzirá o contato com o aparelho reduzindo assim o risco de contaminações vindas do aparelho para o alimento por meio das mãos do manipulador, além de reduzir possíveis erros.

As funcionalidades do sistema implementadas foram: consulta de agenda, agendar, pesquisar, excluir e finalizar os pedidos. Para além disso, foi adicionado a função adicionar e excluir ingredientes do estoque, marcar o tempo de um produto no forno, consultar o tempo restante, a temperatura atual do forno e ligar e desligar ele.

A modelagem do *design* de interação com o usuário permitiu a interação mais prática com o sistema utilizando a voz, criando palavras e frases para que o usuário possa se comunicar com o sistema através da implementação de uma *skill* e se tornou capaz de executar as funcionalidades.

Além disso, ele passou por um teste funcional e por um cálculo de latência que mostraram erros que foram corrigidos e valores de latência aceitáveis, tornando assim um sistema válido e o aplicativo sendo executado com suas funcionalidades de forma correta

Quanto ao tempo do comando de voz, os estudos colocam esse prazo para aplicativos de computador e não para comandos de voz. Contudo, se levarmos em consideração que 10 segundos é o limite para o usuário manter a atenção no diálogo e 100 milissegundos o é o limite para que o usuário sinta que o sistema está em sua funcionalidade plena, é possível concluir que o sistema de voz pode ter um tempo de resposta aceitável, mesmo que em algum momento possa perceber um atraso, visto que o seu tempo máximo foi 1.5 s e a maioria foi abaixo de 100 ms.

Para trabalhos futuros, seria interessante ainda adicionar lembretes através da Alexa, para notificar que o produto está pronto para retirar do forno e lembrar dos pedidos pelo menos uma hora antes do horário de entrega. Além disso, ter a opção de adicionar mais produtos em um só pedido.

É importante também realizar um teste de observação com usuários reais para verificar se o sistema realmente seria útil a eles, além de analisar a usabilidade, que envolve a

eficácia, a eficiência e satisfação dos usuários.

## REFERÊNCIAS

ABRAMS, B. **Google Assistente**: tornando sua rotina mais simples. [S. l.], 2018. Disponível em: <https://www.thinkwithgoogle.com/intl/pt-br/estrat%C3%A9gias-de-marketing/apps-e-mobile/google-assistente-tornando-sua-rotina-mais-simples/>. Acesso em: 28 set. 2020.

AMAZON ALEXA. **Alexa**. c2020. Disponível em: <https://developer.amazon.com/pt-BR/alexa>. Acesso em: 18 de ago. 2020.

AMAZON.COM, I. **Alexa Design Guide**. [S. l.], c2022a. Disponível em: <https://developer.amazon.com/en-US/docs/alexa/alexa-design/get-started.html>. Acesso em: 27 nov. 2022.

AMAZON.COM, I. **O que são as skills da Alexa?** [S. l.], c2022b. Disponível em: <https://developer.amazon.com/en-US/docs/alexa/alexa-design/get-started.html>. Acesso em: 27 nov. 2022.

APPLE. **Siri**. c2020. Disponível em: <https://www.apple.com/br/siri/>. Acesso em: 18 de ago. 2020.

ARAÚJO JÚNIOR, A. P. de; CHAGAS, C. V. das; FERNANDES, R. G. Uma rápida análise sobre automação industrial. **Redes para Automação Industrial**, Natal, 2003.

BARROS, T. T. **Sistema de Gestão para Microempreendedores**. Monografia (Especialização em Engenharia de Software) – Universidade Federal do Paraná, 2015. Disponível em: <https://acervodigital.ufpr.br/bitstream/handle/1884/39529/R%20-%20E%20-%20THIAGO%20TANIMOTO%20BARROS.pdf?sequence=2&isAllowed=y>. Acesso em: 28 set. 2020.

BATISTA, G. A. **Sistema de Controle de Pedidos via Web Padaria Fácil**. Monografia (Graduação em Análise e Desenvolvimento de Sistemas) – Instituto Municipal de Ensino Superior de Assis, 2015. Disponível em: <https://cepein.femanet.com.br/BDigital/arqTccs/1211320461.pdf>. Acesso em: 28 set. 2020.

BAYER, F. M.; ECKHARDT, M.; MACHADO, R.

**Automação de Sistemas** – Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2011.

BICUDO, L. **O que são e para que servem os alto-falantes inteligentes?** 2019. Disponível em: <https://gizmodo.uol.com.br/o-que-sao-e-para-que-servem-os-alto-falantes-inteligentes/>. Acesso em: 27 set. 2020.

CAMPBELL, S. **Make an arduino temperature sensor (thermistor tutorial)**. [S. l.], c2022. Disponível em: <https://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/#:~:text=The%20value%20of%20the%20resistor,a%20multimeter%20to%20find%20out>. Acesso em: 28 nov. 2022.

CHIARADIA, T. S.; SEABRA, R. D.; MATTEDI, A. P. Avaliação da usabilidade do assistente virtual siri: Um estudo de caso com usuários jovens e idosos. **Informática na Educação: teoria prática**, Porto Alegre, 2017. Disponível em: <https://seer.ufrgs.br/index.php/InfEducTeoriaPratica/article/view/69345>. Acesso em: 27 nov. 2022.

COELHO, I. **Qual módulo WiFi ESP32 é ideal para meu projeto?** c2022. Disponível em: <https://www.filipeflop.com/blog/qual-modulo-wifi-esp32-e-ideal-para-meu-projeto/>. Acesso em: 27 nov. 2022.

DELAMARO, M. **Introdução ao Teste de Software**. Rio de Janeiro: Grupo GEN, 2016.

DIÁRIO DO COMÉRCIO. **Segmento da confeitaria detém 25% do setor de panificação**. [S. l.], 2020. Disponível em: <https://diariodocomercio.com.br/negocios/segmento-da-confeitaria-detem-25-do-setor-de-panificacao>. Acesso em: 8 de ago. 2020.

ELETROGATE. **Termistor NTC para Controle de Temperatura**. [S. l.], 2022. Disponível em: <https://blog.eletrogate.com/termistor-ntc-para-controle-de-temperatura/>. Acesso em: 28 nov. 2022.

FABIÃO, R. **Como se tornar um Microempreendedor Individual MEI**. [S. l.], 20–. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/ufs/ap/artigos/como-se-tornar-um-microempreendedor-individual-mei, b66180656e7f0510VgnVCM1000004c00210aRCRD#:~:text=as%20duas%20atividades%3B-,Como%20se%20inscrever,DAS%2C%20declara%C3%A7%C3%A3o%20anual%20ou%20baixa>. Acesso em: 8 de ago. 2020.

FERNANDES, S. R. L. **"Alexa, como vai minha plantinha?"**: Sistema de monitoramento inteligente utilizando assistente virtual para auxiliar no cultivo caseiro de hortaliças. Monografia (Graduação em Engenharia de Computação) – Universidade Federal do Ceará, 2021.

FINNEGAN, M. **Oracle lança assistente de voz para aplicativos de negócios**. [S. l.], 2019. Disponível em: <https://computerworld.com.br/2019/09/20/oracle-lanca-assistente-de-voz-para-aplicativos-de-negocios/>. Acesso em: 8 de ago. 2020.

GOGONI, R. **O que é a Alexa? [ou melhor, quem é]**. [S. l.], c2020. Disponível em: <https://tecnoblog.net/295738/o-que-e-a-alexa-ou-melhor-quem-e/#:~:text=Assim%20como%20suas%20concorrentes%20a,de%20m%C3%BAasicas%20ou%20reproduzir%20podcasts>. Acesso em: 20 set. 2020.

GOOGLE. **Google Assistant**. c2020. Disponível em: [https://assistant.google.com/intl/pt\\_br/](https://assistant.google.com/intl/pt_br/). Acesso em: 28 set. 2020.

GS1 BRASIL. **Estudo de automação do mercado brasileiro: Os impactos da automação entre empresas e consumidores no brasil**. [S. l.], 2017. Disponível em: [https://www.gs1br.org/educacao-e-eventos/EstudoOnda5/201703\\_Estudo\\_de\\_Automa%C3%A7%C3%A3o\\_GS1Brasil.pdf](https://www.gs1br.org/educacao-e-eventos/EstudoOnda5/201703_Estudo_de_Automa%C3%A7%C3%A3o_GS1Brasil.pdf). Acesso em: 26 set. 2022.

GS1 BRASIL. **Índice de Automação do Mercado Brasileiro e de Consumidores**. [S. l.], 2020. Disponível em: <https://www.gs1br.org/indices-e-pesquisas/estudos-e-pesquisas/Documents/Diagn%C3%B3stico%20de%20Automa%C3%A7%C3%A3o%20-%20Setembro%202020.pdf>. Acesso em: 27 set. 2022.

GUERRA, O.; TEIXEIRA, F. A sobrevivência das pequenas empresas no desenvolvimento capitalista. In: **Brazilian Journal of Political Economy**. [S. n.], 2010. v. 30, p. 124–139. Disponível em: <https://www.scielo.br/j/rep/a/XSHRGfXzPy6TRZtkyWBXtXG/?lang=pt>. Acesso em: 28 set. 2022.

- GUTIÉRREZ, J.; ESCALONA, M.; MEJÍAS, M.; TORRES, J. Generating test cases from sequences of use cases. In: **WEBIST 2006: 2nd International Conference on Web Information Systems and Technologies**. [S. l.: s. n.], 2006. p. 473–476.
- HEUMANN, J. Generating test cases from sequences of use case. In: **Second International Conference on Web Information Systems and Technologies**. [S. l.: s. n.], 2001.
- HOME CONNECT GMBH. **Home Connect Oven**. c2022. Disponível em: <https://www.amazon.ca/Home-Connect-GmbH-Oven/dp/B07W94PNWJ>. Acesso em: 1 set. 2020.
- JORNAL CONTÁBEL REDE. **5 dicas de gestão para mei**. [S. l.], 2019. Disponível em: <https://www.jornalcontabil.com.br/5-dicas-de-gestao-para-mei/>. Acesso em: 8 de ago. 2020.
- LEAL, P. **1,7 milhão de famílias brasileiras dependem da renda de um micro-empendedor individual**. [S. l.], 2021. Disponível em: <https://ocp.news/economia/17-milhao-de-familias-brasileiras-dependem-da-renda-de-um-microempendedor-individual>. Acesso em: 8 de ago. 2020.
- LIMA, G. **MongoDB: o banco baseado em documentos**. [S. l.], 2021. Disponível em: <https://www.alura.com.br/artigos/mongodb-o-banco-baseado-em-documentos>. Acesso em: 27 nov. 2022.
- LOBO, E. J. R. **Guia prático de engenharia de software**. São Paulo: Digerati Books, 2009.
- LUÍS, V.; CAMARGO, A. de. **Guia prático de engenharia de software**. São Paulo: Érica, 2013.
- MACEDO, A. L.; LORENZATTI, A.; REATEGUI, E. B.; BEHAR, P. A. Um estudo sobre a introdução de um assistente virtual para suporte à escrita coletiva. In: **RENTE: revista novas tecnologias na educação**. Porto Alegre: [S. n.], 2007.
- MALINE, L. **Amazon Alexa chega ao Brasil totalmente adaptada para o português**. [S. l.], 2021. Disponível em: <https://www.techtudo.com.br/noticias/2019/10/amazon-alexa-chega-ao-brasil-totalmente-adaptada-para-o-portugues.ghtml>. Acesso em: 18 de ago. 2020.
- MARTINS, J. C. C. **Técnicas Para Gerenciamento de Projetos de Software**. Rio de Janeiro: Brasport, 2007.
- MAUÉS, M. P. **Um olhar sobre os assistentes virtuais personificados e a voz como interface**. Dissertação (Mestrado em Design) – Pontífica Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2019.
- MAYER, L. **HOMESTOK**. 2021. Disponível em: <http://minutoligado.com.br/noticias/saiba-mais-sobre-o-mercado-de-doces-no-brasil/>. Acesso em: 28 set. 2022.
- MOL, B. **Google Assistant: como funciona a assistente virtual do google**. [S. l.], 2020. Disponível em: <https://take.net/blog/inovacao/google-assistant>. Acesso em: 28 de set. 2020.
- MUNDO PROJETADO. **Termistor NTC com Arduino**. [S. l.], c2022. Disponível em: <https://mundoprojetado.com.br/termistor-ntc/>. Acesso em: 28 nov. 2022.
- NETO, A.; CLAUDIO, D. Introdução a teste de software. In: **Engenharia de Software Magazine**. [S. l.: s. n.], 2007. v. 1, p. 22.

NIELSEN, J. **Tempos de Resposta**: Os 3 limites importantes. c2022. Disponível em: <https://www.nngroup.com/articles/response-times-3-important-limits/>. Acesso em: 1 set. 2020.

OLIVEIRA, C. **Como o Google quer usar o assistente de voz para ajudar pequenos negócios**. [S. l.], 2019. Disponível em: <https://revistapegn.globo.com/Tecnologia/noticia/2019/02/como-o-google-quer-usar-o-assistente-de-voz-para-ajudar-pequenos-negocios.html#:~:text=Uma%20tend%C3%Aancia%20chamada%20de%20hands,acontecem%20sem%20encostar%20no%20celular.&text=Lauren%20aponta%20que%20as%20marcas,a%20onda%20da%20intelig%C3%Aancia%20artificial>. Acesso em: 18 de ago. 2020.

OLIVEIRA, E. **Como usar com Arduino – Termistor NTC 10K (Sensor de Temperatura)**. [S. l.], c2022. Disponível em: <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-termistor-ntc-10k-sensor-de-temperatura>. Acesso em: 27 nov. 2022.

OLIVEIRA, J. **Alexa, quais marcas já criaram skills no Brasil?**: Serviço de voz da amazon tem mais de 300 aplicativos em português. [S. l.], 2019. Disponível em: <https://propmark.com.br/mercado/alexa-quais-marcas-ja-criaram-skills-no-brasil/>. Acesso em: 20 set. 2020.

PEREIRA, R. de S. **SISCONT**: sistema de automatização de forno elétrico via dispositivo móvel. Monografia (Especialização em Automação e Controle de Processos Industriais) – Universidade Tecnológica Federal do Paraná, Ponta Grossa,, 2017.

PORTAL DO EMPREENDEDOR. **O que é ser um MEI?** [S. l.], c2020. Disponível em: <http://www.portaldoempreendedor.gov.br/temas/quero-ser/formalize-se/O-que-e-ser-um-mei>. Acesso em: 20 set. 2020.

PRISCILA. **Docerias, bolerias e confeitarias gourmet**: oportunidades de negócio que dão água na boca. [S. l.], 2019. Disponível em: <https://blog.sebrae-sc.com.br/docerias-bolerias-e-confeitarias-gourmet-oportunidades-de-negocio-que-dao-agua-na-boca/>. Acesso em: 8 de ago. 2020.

REDAÇÃO. **Conheça as atividades mais procuradas por quem quer ser MEI**. [S. l.], 2020. Disponível em: <https://www.jornalcontabil.com.br/conheca-as-atividades-mais-procuradas-por-quem-quer-ser-mei/>. Acesso em: 24 set. 2020.

RIOS, E.; MOREIRA, T. **Teste de Software**. Rio de Janeiro: Alta Books, 2013.

RODRIGUES, J. P. F. **Assistente Virtual para Fatura Inteligente**. Dissertação (Mestrado em Engenharia Informática com especialização em Engenharia de Software) – Universidade de Lisboa, 2019.

ROGGIA, L.; FUENTES, R. C.

**Automação Industrial** – Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2016.

ROTEAMENTO. **Express**. [S. l.], c2022. Disponível em: <https://expressjs.com/pt-br/guide/routing.html>. Acesso em: 27 nov. 2022.

SEBRAE. **Tendências de alimentos e bebidas para MEIs em 2022**. [S. l.], 2022a. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/tendencias-de-alimentos-e-bebidas-para-meis-em-2022,5eef9ac095d8f710VgnVCM100000d701210aRCRD>. Acesso em: 28 set. 2022.

SEBRAE. **Tudo o que você precisa saber sobre o MEI**. [S. l.], 2022b. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/tudo-o-que-voce-precisa-saber-sobre-o-mei,caa7d72060589710VgnVCM100000d701210aRCRD#:~:text=MEI%20significa%20Microempreendedor%20Individual%2C%20ou,direitos%20de%20uma%20pessoa%20jur%C3%ADica>. Acesso em: 27 set. 2022.

SEBRAE. **Como montar um serviço de automação comercial**. [S. l.], c2022. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/ideias/como-montar-um-servico-de-automacao-comercial,a8397a51b9105410VgnVCM1000003b74010aRCRD>. Acesso em: 24 ago. 2020.

SEBRAE; ABIP; ITPC. Perfil do microempreendedor individual de alimentos e bebidas no brasil: Indicadores, desafios e tendências. In: **Empreender em alimentos e bebidas**. [S. l.: s. n.], 2017. v. 1.

SEBRAE (CE). **Pequenos Negócios ajudam a movimentar economia brasileira**. [S. l.], 2019. Disponível em: <https://diariodonordeste.verdesmares.com.br/negocios/pequenos-negocios-ajudam-a-movimentar-economia-brasileira-1.2159231>. Acesso em: 8 de ago. 2020.

SEBRAE (ES). **Automação Comercial: o que é e quais os benefícios para os pequenos negócios**. [S. l.], 2019. Disponível em: <https://blog.sebraees.com.br/2019/08/19/automacao-comercial-o-que-e-e-quais-os-beneficios-para-os-pequenos-negocios>. Acesso em: 24 ago. 2020.

SHARMA, P. **Group Tasks Manager**. c2022. Disponível em: <https://www.amazon.com/Piyush-Sharma-Group-Tasks-Manager/dp/B07FYH2VCG>. Acesso em: 1 set. 2020.

SILVA, E. E. **Como fazer um plano de testes baseado em casos de uso**. [S. l.], c2022. Disponível em: <http://www.linhadecodigo.com.br/artigo/2958/como-fazer-um-plano-de-testes-baseado-em-casos-de-uso.aspx>. Acesso em: 3 dez. 2022.

SILVEIRA, C. B. **O que é Automação Industrial**. [S. l.], c2020. Disponível em: <https://www.citisystems.com.br/o-que-e-automacao-industrial/>. Acesso em: 24 ago. 2020.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson Prentice Hall, 2011. v. 9.

SOUZA, E. T. de. **Vocal: assistente para o uso de smartphones operado por voz**. Monografia (Trabalho de conclusão de curso (Graduação em Ciência da Computação)) – Universidade Regional de Blumenau, 2016.

TECHTUTORIALSX. **ESP32: solicitações http get**. [S. l.], 2022. Disponível em: <https://suadica.com/dica.php?d=432&t=solicitacoes-http-post-no-microcontrolador-esp32>. Acesso em: 27 nov. 2022.

THAKUR, A. **A Conversation UI experience with SiriKit**. [S. l.], 2019. Disponível em: <https://medium.com/@i.m.amit.k.thakur/a-conversation-ui-experience-with-sirikit-f46b8b03da45>. Acesso em: 27 set. 2020.

TORRES, V. **Socorro: o que é a ide do arduino?!** [S. l.], 2013. Disponível em: <http://www.natalnet.br/ura/?p=438>. Acesso em: 27 nov. 2022.

TOTAL COMMERCE SOLUTIONS. **Oracle lança assistente de voz para aplicativos de negócios**. [S. l.], c2020. Disponível em: <https://www.totall.com.br/oracle-lanca-assistente-de-voz-para-aplicativos-de-negocios/>. Acesso em: 18 de ago. 2020.

TURTON, R. **SiriKit Tutorial for iOS**. [S. l.], 2017. Disponível em: <https://www.raywenderlich.com/600-sirikit-tutorial-for-ios>. Acesso em: 27 set. 2020.

WHIRLPOOL CORPORATION. **KitchenAid Connected Ovens**. c2022. Disponível em: <https://www.amazon.com/WHIRLPOOL-GLOBAL-HEADQUARTERS-KitchenAid-Connected/dp/B07RKY87R6>. Acesso em: 1 set. 2020.

XIAOQUANG, L.; YONG, Y.; JU1, Z. Design and implementation of smart cooking based on amazon echo. In: **IJSCAI: International Journal of Smart Computing and Artificial Intelligence**. [S. l.: s. n.], 2018. p. 37–45.

## APÊNDICE A – TABELAS DOS CASOS DE TESTE

Tabela 20 – Caso de teste para o caso de uso agendar pedido

Caso de uso: Agendar pedido	
Fluxo básico - <i>Slots</i> não foram preenchidos	<ol style="list-style-type: none"> <li>1. Usuário pede para agendar um pedido - Quero agendar um pedido</li> <li>2. Alexa pergunta de quem é o pedido - Tudo bem, qual é o nome do cliente do pedido?</li> <li>3. Usuário responde o nome do cliente - {nome do cliente}</li> <li>4. Alexa pergunta para que dia é o pedido - Para que dia é o pedido?</li> <li>5. Usuário responde o dia - {data}</li> <li>6. Alexa pergunta qual é o horário de entrega - Qual é o horário de entrega?</li> <li>7. Alexa pergunta qual é o produto - Qual é o produto do pedido?</li> <li>8. Usuário responde o produto - {produto}</li> <li>9. Alexa pergunta qual é a quantidade - Qual é a quantidade?</li> <li>10. Usuário responde a quantidade - {quantidade}</li> <li>11. Alexa responde que pedido foi agendado - Pronto, pedido de {nome do cliente} com {quantidade} {produto} foi agendado para o dia {data} às {hora} horas</li> </ol>

*Continua na próxima página*

Tabela 20 – Continuação da tabela

Caso de uso: Agendar pedido	
Fluxo alternativo 2 - <i>Slots</i> {data}, {hora}, {produto} e {quantidade} NÃO foram preenchidos	<ol style="list-style-type: none"> <li>1. Usuário pede para agendar um pedido - Quero agendar o pedido de {nome do cliente}</li> <li>2. Alexa pergunta para que dia é o pedido - Para que dia é o pedido?</li> <li>3. Usuário responde o dia - {data}</li> <li>4. Alexa pergunta qual é o horário de entrega - Qual é o horário de entrega?</li> <li>5. Usuário responde o dia - {hora}</li> <li>6. Alexa pergunta qual é o produto - Qual é o produto do pedido?</li> <li>7. Usuário responde o produto - {produto}</li> <li>8. Alexa pergunta qual é a quantidade - Qual é a quantidade?</li> <li>9. Usuário responde a quantidade - {quantidade}</li> <li>10. Alexa responde que pedido foi agendado - Pronto, pedido de {nome do cliente} com {quantidade} {produto} foi agendado para o dia {data} às {hora} horas</li> </ol>

Fonte: elaborada pela autora.

Tabela 21 – Caso de teste para o caso de uso excluir pedido

Caso de uso: Excluir pedido	
Fluxo básico - Slot {nome do cliente} está preenchido e existe só 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir o pedido de um cliente <ul style="list-style-type: none"> <li>- Quero excluir o pedido de {nome do cliente}</li> </ul> </li> <li>2. Alexa responde que o pedido foi excluído: <ul style="list-style-type: none"> <li>- Pedido com {quantidade} {nome do produto} para o dia {data} foi excluído</li> </ul> </li> </ol>
Fluxo alternativo 1 - Slot {nome do cliente} NÃO está preenchido e existe só 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir o pedido de um cliente <ul style="list-style-type: none"> <li>- Quero excluir um pedido</li> </ul> </li> <li>2. Alexa pergunta de quem é o pedido <ul style="list-style-type: none"> <li>- Tudo bem, de qual cliente você quer excluir o pedido?</li> </ul> </li> <li>3. Usuário responde o nome do cliente <ul style="list-style-type: none"> <li>- {nome do cliente}</li> </ul> </li> <li>4. Alexa responde que o pedido foi excluído: <ul style="list-style-type: none"> <li>- Pedido com {quantidade} {nome do produto} para o dia {data} foi excluído</li> </ul> </li> </ol>
Fluxo alternativo 2 - Slot {nome do cliente} está preenchidos e existem MAIS de 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para finalizar o pedido de um cliente <ul style="list-style-type: none"> <li>- Finalize o pedido de {nome do cliente}</li> </ul> </li> <li>2. Alexa responde os pedidos agendados do cliente <ul style="list-style-type: none"> <li>- {nome do cliente} tem {quantidade de pedidos} pedidos que são: pedido {numero do pedido}: {quantidade} {produto} para o dia {data} às {hora} horas]. Qual deseja finalizar?</li> </ul> </li> <li>3. Usuário responde o número do pedido <ul style="list-style-type: none"> <li>- {número do pedido}</li> </ul> </li> <li>4. Alexa responde que o pedido foi finalizado: <ul style="list-style-type: none"> <li>- Pedido com {quantidade} {nome do produto} para o dia {data} foi finalizado</li> </ul> </li> </ol>

*Continua na próxima página*

Tabela 21 – *Continuação da tabela*

Caso de uso: Excluir pedido	
Fluxo alternativo 3 - Slot {nome do cliente} está preenchidos e existem MAIS de 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir o pedido de um cliente - Quero excluir o pedido de {nome do cliente}</li> <li>2. Alexa responde os pedidos agendados do cliente - {nome do cliente} tem {quantidade de pedidos} pedidos que são: pedido {numero do pedido}: {quantidade} {produto} para o dia {data} às {hora} horas]. Qual deseja excluir?</li> <li>3. Usuário responde o número do pedido - {número do pedido}</li> <li>4. Alexa responde que o pedido foi excluído: - Pedido com {quantidade} {nome do produto} para o dia {data} foi excluído</li> </ol>

*Continua na próxima página*

Tabela 21 – *Continuação da tabela*

Caso de uso: Excluir pedido	
Fluxo alternativo 4 - Slot {nome do cliente} NÃO está preenchidos e existe MAIS de 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir o pedido de um cliente <ul style="list-style-type: none"> <li>- Quero excluir um pedido</li> </ul> </li> <li>2. Alexa pergunta de quem é o pedido <ul style="list-style-type: none"> <li>- Tudo bem, de qual cliente você quer excluir o pedido?</li> </ul> </li> <li>3. Usuário responde o nome do cliente <ul style="list-style-type: none"> <li>- {nome do cliente}</li> </ul> </li> <li>4. Alexa responde os pedidos agendados do cliente <ul style="list-style-type: none"> <li>- {nome do cliente} tem {quantidade de pedidos} pedidos que são: pedido {numero do pedido}: {quantidade} {produto} para o dia {data} às {hora} horas]. Qual deseja excluir?</li> </ul> </li> <li>5. Usuário responde o número do pedido <ul style="list-style-type: none"> <li>- {número do pedido}</li> </ul> </li> <li>6. Alexa responde que o pedido foi excluído: <ul style="list-style-type: none"> <li>- Pedido com {quantidade} {nome do produto} para o dia {data} foi excluído.</li> </ul> </li> </ol>

Fonte: elaborada pela autora.

Tabela 22 – Caso de teste para o caso de uso excluir ingrediente do estoque

Caso de uso: Excluir ingrediente do estoque	
Fluxo básico - Slots {nome do ingrediente} e {quantidade} preenchidos e tem este ingrediente no estoque	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir um ingrediente <ul style="list-style-type: none"> <li>- Exclua {quantidade} {nome do ingrediente} do estoque</li> </ul> </li> <li>2. Alexa responde que ingrediente foi excluído do estoque: <ul style="list-style-type: none"> <li>- {quantidade} {nome do ingrediente} foi excluído do estoque</li> </ul> </li> </ol>
Fluxo alternativo 1 - Slots {nome do ingrediente} e {quantidade} preenchidos e NÃO tem este ingrediente no estoque	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir um ingrediente <ul style="list-style-type: none"> <li>- Tem {nome do ingrediente} no estoque?</li> </ul> </li> <li>2. Alexa responde que não tem este ingrediente no estoque: <ul style="list-style-type: none"> <li>- Não tem {nome do ingrediente} no estoque</li> </ul> </li> </ol>
Fluxo alternativo 2 - Slot {nome do ingrediente} preenchido e tem este ingrediente no estoque	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir um ingrediente <ul style="list-style-type: none"> <li>- Exclua {nome do ingrediente} do estoque</li> </ul> </li> <li>2. Alexa pergunta qual é a quantidade de ingrediente que o usuário quer excluir: <ul style="list-style-type: none"> <li>- Qual é a quantidade de {ingrediente} que você quer excluir?</li> </ul> </li> <li>3. Usuário responde a quantidade <ul style="list-style-type: none"> <li>- {quantidade}</li> </ul> </li> <li>4. Alexa responde que ingrediente foi excluído do estoque: <ul style="list-style-type: none"> <li>- {quantidade} {nome do ingrediente} foi excluído do estoque e restaram {quantidade restante}</li> </ul> </li> </ol>

*Continua na próxima página*

Tabela 22 – *Continuação da tabela*

Caso de uso: Excluir ingrediente do estoque	
Fluxo alternativo 3 - Slot {nome do ingrediente} preenchido e NÃO tem este ingrediente no estoque	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir um ingrediente <ul style="list-style-type: none"> <li>- Exclua {nome do ingrediente} do estoque</li> </ul> </li> <li>2. Alexa pergunta qual é a quantidade de ingrediente que o usuário quer excluir: <ul style="list-style-type: none"> <li>- Qual é a quantidade de {ingrediente} que você quer excluir?</li> </ul> </li> <li>3. Usuário responde a quantidade <ul style="list-style-type: none"> <li>- {quantidade}</li> </ul> </li> <li>4. Alexa responde que não tem este ingrediente no estoque: <ul style="list-style-type: none"> <li>- Não tem {nome do ingrediente} no estoque e restaram {quantidade restante}</li> </ul> </li> </ol>
Fluxo alternativo 4 - Slot {nome do ingrediente} NÃO foi preenchido e tem este ingrediente no estoque	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir um ingrediente <ul style="list-style-type: none"> <li>- Quero excluir um ingrediente do estoque</li> </ul> </li> <li>2. Alexa pergunta qual é o ingrediente que o usuário quer excluir: <ul style="list-style-type: none"> <li>- Certo, qual é o ingrediente que você quer excluir?</li> </ul> </li> <li>3. Usuário responde o ingrediente <ul style="list-style-type: none"> <li>- {ingrediente}</li> </ul> </li> <li>4. Alexa pergunta qual é a quantidade de ingrediente que o usuário quer excluir: <ul style="list-style-type: none"> <li>- Qual é a quantidade de {ingrediente} que você quer excluir?</li> </ul> </li> <li>5. Usuário responde a quantidade <ul style="list-style-type: none"> <li>- {quantidade}</li> </ul> </li> <li>6. Alexa responde que ingrediente foi excluído do estoque: <ul style="list-style-type: none"> <li>- {quantidade} {nome do ingrediente} foi excluído do estoque e restaram {quantidade restante}</li> </ul> </li> </ol>

*Continua na próxima página*

Tabela 22 – *Continuação da tabela*

Caso de uso: Excluir ingrediente do estoque	
Fluxo alternativo 5 - Slot {nome do ingrediente} NÃO foi preenchido e NÃO tem este ingrediente no estoque	<ol style="list-style-type: none"> <li>1. Usuário pede para excluir um ingrediente - Quero excluir um ingrediente do estoque</li> <li>2. Alexa pergunta qual é o ingrediente que o usuário quer excluir: - Certo, qual é o ingrediente que você quer excluir?</li> <li>3. Usuário responde o ingrediente - {ingrediente}</li> <li>4. Alexa pergunta qual é a quantidade de ingrediente que o usuário quer excluir: - Qual é a quantidade de {ingrediente} que você quer excluir?</li> <li>5. Usuário responde a quantidade - {quantidade}</li> <li>6. Alexa responde que não tem este ingrediente no estoque: - Não tem {nome do ingrediente} no estoque e restaram {quantidade restante}</li> </ol>

Fonte: elaborada pela autora.

Tabela 23 – Caso de teste para o caso de uso pesquisar pedidos do cliente

Caso de uso: Pesquisar pedidos do cliente	
Fluxo básico - Slot {nome do cliente} preenchido e cliente tem pedidos agendados	<p>1. Usuário pede os pedidos de um cliente: - Quero saber quais são os pedidos de {nome do cliente}</p> <p>2. Alexa responde os pedidos agendados do cliente - Os pedidos de {nome do cliente} são: [{quantidade} {produto} para {hora} horas]</p>
Fluxo alternativo 1 - Slot {nome do cliente} preenchidos e cliente NÃO tem pedidos agendados	<p>1. Usuário pede os pedidos de um cliente: Quero saber quais são os pedidos de {nome do cliente}</p> <p>2. Alexa responde que não tem pedidos agendados para o cliente{nome do cliente} não tem pedidos agendados</p>
Fluxo alternativo 2 - Slots {nome do cliente} e {data} preenchidos e cliente tem pedidos agendados para esta data	<p>1. Usuário pede os pedidos de um cliente: Quero saber quais são os pedidos de {nome do cliente} do dia {data}</p> <p>2. Alexa os pedidos agendados do cliente para este dia - Os pedidos de maria do dia {data} são: [{quantidade} {produto} para {hora} horas]</p>
Fluxo alternativo 3 - Slots {nome do cliente} e {data} preenchidos e cliente NÃO tem pedidos agendados para esta data	<p>1. Usuário pede os pedidos de um cliente: - Quero saber quais são os pedidos de {nome do cliente} do dia {data}</p> <p>2. Alexa os pedidos agendados do cliente para este dia {nome do cliente} não tem pedidos para o dia {dia}</p>
Fluxo alternativo 4 - Slots {nome do cliente} e {status do pedido} preenchidos e cliente tem pedidos para este cliente com este status	<p>1. Usuário pede os pedidos de um cliente: - Quero saber quais são os pedidos de {nome do cliente} {status do pedido}?</p> <p>2. Alexa os pedidos agendados do cliente para este dia - Os pedidos de {nome do cliente} {status do pedido} são: [{quantidade} {produto} para {hora} horas]</p>
Fluxo alternativo 5 - Slot {nome do cliente} e {status do pedido} preenchidos e cliente tem pedidos para este cliente com este status	<p>1. Usuário pede os pedidos de um cliente: - Quero saber quais são os pedidos de {nome do cliente} {status do pedido}?</p> <p>2. Alexa os pedidos agendados do cliente para este dia - {nome do cliente} não tem pedidos {status do pedido}</p>

Fonte: elaborada pela autora.

Tabela 24 – Caso de teste para o caso de uso finalizar pedido

Caso de uso: Finalizar pedido	
Fluxo básico - Slot {nome do cliente} está preenchido e existe só 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para finalizar o pedido de um cliente <ul style="list-style-type: none"> <li>- Finalize o pedido de {nome do cliente}</li> </ul> </li> <li>2. Alexa responde que o pedido foi finalizado: Pedido com {quantidade} {nome do produto} para o dia {data} foi finalizado</li> </ol>
Fluxo alternativo 1 - Slot {nome do cliente} NÃO está preenchido e existe só 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para finalizar o pedido de um cliente <ul style="list-style-type: none"> <li>- Finalize um pedido</li> </ul> </li> <li>2. Alexa pergunta de quem é o pedido <ul style="list-style-type: none"> <li>- Tudo bem, de qual cliente você quer finalizar o pedido?</li> </ul> </li> <li>3. Usuário responde o nome do cliente <ul style="list-style-type: none"> <li>- {nome do cliente}</li> </ul> </li> <li>4. Alexa responde que o pedido foi finalizado: <ul style="list-style-type: none"> <li>- Pedido com {quantidade} {nome do produto} para o dia {data} foi finalizado</li> </ul> </li> </ol>
Fluxo alternativo 2 - Slot {nome do cliente} está preenchidos e existem MAIS de 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para finalizar o pedido de um cliente <ul style="list-style-type: none"> <li>- Finalize o pedido de {nome do cliente}</li> </ul> </li> <li>2. Alexa responde os pedidos agendados do cliente <ul style="list-style-type: none"> <li>- {nome do cliente} tem {quantidade de pedidos} pedidos que são: pedido {numero do pedido}: {quantidade} {produto} para o dia {data} às {hora} horas]. Qual deseja finalizar?</li> </ul> </li> <li>3. Usuário responde o número do pedido <ul style="list-style-type: none"> <li>- {número do pedido}</li> </ul> </li> <li>4. Alexa responde que o pedido foi finalizado: <ul style="list-style-type: none"> <li>- Pedido com {quantidade} {nome do produto} para o dia {data} foi finalizado</li> </ul> </li> </ol>
Fluxo alternativo 3 - Slot {nome do cliente} NÃO está preenchidos e existe MAIS de 1 pedido para o cliente	<ol style="list-style-type: none"> <li>1. Usuário pede para finalizar o pedido de um cliente <ul style="list-style-type: none"> <li>- Finalize um pedido</li> </ul> </li> <li>2. Alexa pergunta de quem é o pedido <ul style="list-style-type: none"> <li>- Tudo bem, de qual cliente você quer finalizar o pedido?</li> </ul> </li> <li>3. Usuário responde o nome do cliente <ul style="list-style-type: none"> <li>- {nome do cliente}</li> </ul> </li> <li>4. Alexa responde os pedidos agendados do cliente <ul style="list-style-type: none"> <li>- {nome do cliente} tem {quantidade de pedidos} pedidos que são: pedido {numero do pedido}: {quantidade} {produto} para o dia {data} às {hora} horas]. Qual deseja finalizar?</li> </ul> </li> <li>5. Usuário responde o número do pedido <ul style="list-style-type: none"> <li>- {número do pedido}</li> </ul> </li> <li>6. Alexa responde que o pedido foi finalizado: <ul style="list-style-type: none"> <li>- Pedido com {quantidade} {nome do produto} para o dia {data} foi finalizado.</li> </ul> </li> </ol>

Fonte: elaborada pela autora.

Tabela 25 – Caso de teste para o caso de uso adicionar ingrediente no estoque

Caso de uso: Adicionar ingrediente no estoque	
Fluxo básico - Slots {quantidade} {nome do ingrediente} preenchidos	1. Usuário pede para adicionar ingrediente - Adicione {quantidade} {nome do ingrediente} ao estoque 2. Alexa responde que o ingrediente foi adicionado: - Certo. {quantidade} {nome do ingrediente} foi adicionado ao estoque
Fluxo alternativo 1 - Slots {quantidade} {nome do ingrediente} NÃO foram preenchidos	1. Usuário pede para adicionar ingrediente - Adicione um ingrediente ao estoque 2. Alexa pergunta qual é o ingrediente: - Tudo bem. Qual é o ingrediente que deseja adicionar? 3. Usuário responde o ingrediente - {nome do ingrediente} 4. Alexa pergunta qual é a quantidade: - Tudo bem. Qual é o ingrediente que deseja adicionar? 3. Usuário responde a quantidade - {quantidade} 4. Alexa responde que o ingrediente foi adicionado: - Certo. {quantidade} {nome do ingrediente} foi adicionado ao estoque

Fonte: elaborada pela autora.

Tabela 26 – Caso de teste para o caso de uso ligar passagem de gás para o forno

Caso de uso: Ligar passagem de gás para o forno	
Fluxo básico	1. Usuário pede para ligar o forno - Ligue o forno 2. Alexa responde que o forno foi ligado - O forno está ligado

Fonte: elaborada pela autora.

Tabela 27 – Caso de teste para o caso de uso desligar passagem de gás para o forno

Caso de uso: Desligar passagem de gás para o forno	
Fluxo básico - NÃO tem produtos no forno	1. Usuário pede para desligar o forno - Ligue o forno 2. Alexa responde que o forno foi desligado - O forno está desligado
Fluxo básico - Tem produtos no forno	1. Usuário pede para desligar o forno - Ligue o forno 2. Alexa responde que o forno não foi possível desligar o forno - Não foi possível desligar, pois existem produtos no forno

Fonte: elaborada pela autora.

Tabela 28 – Caso de teste para o caso de uso definir tempo que produto deve ficar no forno

Caso de uso: Definir tempo que produto deve ficar no forno	
Fluxo básico - Slots {nome do produto} e {tempo} preenchidos NÃO tem este produto no forno	1. Usuário pede para marcar tempo que produto deve ficar no forno - Marque {tempo} para assar o {nome do produto} 2. Alexa responde que o forno foi para o forno e quando deve ficar pronto - Pronto. O {nome do produto} foi para o forno às {hora} e deve ficar por {tempo} minutos, finalizando às {hora finalização} horas
Fluxo alternativo 2 - Slots {tempo} NÃO foi preenchido e NÃO tem este produto no forno	1. Usuário avisa que colocou produto no forno Coloquei {nome do produto} no forno 2. Alexa pergunta quanto tempo ele deve ficar lá: - Quanto tempo {nome do produto} deve ficar no forno? 3. Usuário responde o tempo - {tempo} 4. Alexa responde que o forno foi para o forno e quando deve ficar pronto - Pronto. O {nome do produto} foi para o forno às {hora} e deve ficar por {tempo} minutos, finalizando às {hora finalização} horas

Fonte: elaborada pela autora.

Tabela 29 – Caso de teste para o caso de uso consultar quanto tempo falta até a conclusão

Caso de uso: Consultar quanto tempo falta até a conclusão	
Fluxo básico - Existe apenas um produto deste no forno	<ol style="list-style-type: none"> <li>1. Usuário pergunta quanto tempo falta para retirar produto do forno <ul style="list-style-type: none"> <li>- Quanto tempo falta para tirar {nome do produto} do forno</li> </ul> </li> <li>2. Alexa responde que o forno foi para o forno e quando deve ficar pronto <ul style="list-style-type: none"> <li>- Faltam {tempo} minutos para tirar o {produto} do forno</li> </ul> </li> </ol>
Fluxo alternativo 1 - Existe mais produtos deste no forno	<ol style="list-style-type: none"> <li>1. Usuário pergunta quanto tempo falta para tirar produto do forno <ul style="list-style-type: none"> <li>- Quanto tempo falta para tirar {nome do produto} do forno</li> </ul> </li> <li>2. Alexa responde que o forno foi para o forno e quando deve ficar pronto <ul style="list-style-type: none"> <li>- Existem {quantidade do produto} {produto} no forno, faltando [{tempo}] minutos para tirar do forno</li> </ul> </li> </ol>
Fluxo alternativo 2 - Não existe este produto no forno	<ol style="list-style-type: none"> <li>1. Usuário pergunta quanto tempo falta para tirar produto do forno <ul style="list-style-type: none"> <li>- Quanto tempo falta para tirar {nome do produto} do forno</li> </ul> </li> <li>2. Alexa responde que o forno foi para o forno e quando deve ficar pronto <ul style="list-style-type: none"> <li>- Não tem{produto} no forno</li> </ul> </li> </ol>

Fonte: elaborada pela autora.

Tabela 30 – Caso de teste para o caso de uso consultar quais são os produtos que estão no forno

Caso de uso: Consultar quais são os produtos que estão no forno	
Fluxo básico - Existem produtos no forno	<ol style="list-style-type: none"> <li>1. Usuário pergunta quanto tempo falta para tirar produto do forno <ul style="list-style-type: none"> <li>O que tem no forno?</li> </ul> </li> <li>2. Alexa responde os produtos que estão no forno <ul style="list-style-type: none"> <li>Os produtos que estão no forno são [{nome do produto}]</li> </ul> </li> </ol>
Fluxo alternativo 1 - NÃO existem produtos no forno	<ol style="list-style-type: none"> <li>1. Usuário pergunta quanto tempo falta para tirar produto do forno <ul style="list-style-type: none"> <li>O que tem no forno?</li> </ul> </li> <li>2. Alexa responde os produtos que estão no forno <ul style="list-style-type: none"> <li>Não tem produtos no forno</li> </ul> </li> </ol>
Fluxo alternativo 2 - Não existe este produto no forno	<ol style="list-style-type: none"> <li>1. Usuário pergunta quanto tempo falta para tirar produto do forno <ul style="list-style-type: none"> <li>- Quanto tempo falta para tirar {nome do produto} do forno</li> </ul> </li> <li>2. Alexa responde que o forno foi para o forno e quando deve ficar pronto <ul style="list-style-type: none"> <li>- Não tem{produto} no forno</li> </ul> </li> </ol>

Fonte: elaborada pela autora.

Tabela 31 – Caso de teste para o caso de uso consultar a temperatura atual do forno

Caso de uso: Consultar a temperatura atual do forno	
Fluxo básico - Forno está ligado	<p>1. Usuário pergunta qual é a temperatura atual do forno Qual é a temperatura do forno?</p> <p>2. Alexa responde os produtos que estão no forno A temperatura atual do forno é {temperatura} graus.</p>
Fluxo alternativo 1 - NÃO existem produtos no forno	<p>1. Usuário pergunta qual é a temperatura atual do forno Qual é a temperatura do forno?</p> <p>2. Alexa responde os produtos que estão no forno Forno não está ligado</p>
Fluxo alternativo 2 - Não existe este produto no forno	<p>1. Usuário pergunta quanto tempo falta para tirar produto do forno - Quanto tempo falta para tirar {nome do produto} do forno</p> <p>2. Alexa responde que o forno foi para o forno e quando deve ficar pronto - Não tem{produto} no forno</p>

Fonte: elaborada pela autora.

## APÊNDICE B – TRECHOS DE CÓDIGO

Figura 10 – Trecho de código do servidor *web* com as funções de consultar agenda e agendar pedido.

```

app.get('/ConsultarAgenda', (req, res) => {
  let json = JSON.parse(req.query[0])
  Pedido.find({dia: json.dia, "status": "agendado"}).lean().exec(
    function (e, docs) {
      if(docs === null){
        res.send({code: 0})
      }else{
        res.send(docs)
      }
    }
  )
})
app.post('/AgendarPedido', (req, res) => {
  let json = JSON.parse(req.query[0])
  data = {"nomeCli": json.nomeCli, "hora": json.hora, "dia": json.dia, "produto": json.produto, "quantidade": json.quantidade, "status": json.status};
  var pedido = new Pedido(data);
  pedido.save(function (err, doc) {
    if (err) {
      console.log("Error! " + err.message);
      res.send(err)
    }
    else {
      res.send({code: 1})
    }
  });
});
})

```

Fonte: Elaborada pela autora.

Figura 11 – Trecho de código do *software* embarcado.

```

void loop() {

  if (WiFi.status() == WL_CONNECTED) {
    //Check WiFi connection status

    HTTPClient http;

    http.begin("https://server-projeto.herokuapp.com/getInfosForno"); //Specify destination for HTTP request
    int httpResponseCode = http.GET(); //Send the actual POST request

    if (httpResponseCode > 0) {
      DynamicJsonDocument doc(2048);
      deserializeJson(doc, http.getStream());

      long statusForno = doc["statusForno"].as<long>();
      long statusNotificacao = doc["statusNotificacao"].as<long>();

      ligaDesligaForno(statusForno);

      notificar(statusNotificacao);

      if(statusForno == 1){
        enviarValorTemperatura();
      }
    } else {

      Serial.print("Error on sending POST: ");
      Serial.println(httpResponseCode);

    }

    http.end(); //Free resources
  }
}

```

Fonte: Elaborada pela autora.

Figura 12 – Trecho de código do *software* com o método *ligaEDesligaaForno()*

```

void ligaDesligaForno(int statusForno) {

    if (statusForno == 1) {
        digitalWrite(LED_FORNO_DESLIGADO, LOW);
        digitalWrite(LED_FORNO_LIGADO, HIGH);
        digitalWrite(LED_LUZ_FORNO, HIGH);
    } else {
        digitalWrite(LED_LUZ_FORNO, LOW);
        digitalWrite(LED_FORNO_LIGADO, LOW);
        digitalWrite(LED_FORNO_DESLIGADO, HIGH);
    }
}
}

```

Fonte: Elaborada pela autora.

Figura 13 – Trecho de código do *software* com o método *enviaValorTemperatura()*

```

void enviarValorTemperatura() {

    int temperatura = (int) getTemp(pinNTC);
    HTTPClient http;
    String queryString = String("?temperatura=") + String(temperatura);

    http.begin("https://server-projeto.herokuapp.com/setInfosForno" + queryString);

    int httpResponseCode = http.GET(); //Send the actual POST request

    if (httpResponseCode > 0) {

    } else {
        Serial.print("Error on sending POST: ");
        Serial.println(httpResponseCode);
    }

    http.end(); //Free resources

}

```

Fonte: Elaborada pela autora.