

UNIVERSIDADE DO OESTE DE SANTA CATARINA

KENNILDO BASTOS GONÇALVES

INTEGRAÇÃO DE ASSISTENTE VIRTUAL ALEXA COM APLICAÇÃO  
DE GESTÃO APÍCOLA

CHAPECÓ, SC

2025

KENNILDO BASTOS GONÇALVES

INTEGRAÇÃO DE ASSISTENTE VIRTUAL ALEXA COM APLICAÇÃO DE  
GESTÃO APÍCOLA

Projeto de Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação, Área das Ciências Exatas e da Terra, da Universidade do Oeste de Santa Catarina – Unoesc Campus de Chapecó, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Chapecó, SC

2025

## SUMÁRIO

## LISTA DE ILUSTRAÇÕES

Ilustração 1 - Arquitetura do sistema Amazon Alexa Echo Dot .....	45
Ilustração 2 - Modelo Cascata .....	56
Ilustração 3 - Tela de criação de Skill Alexa .....	83
Ilustração 4 - Tipos de linguagem aceitas pela Skill .....	92
Ilustração 5 - Tela para testar Skill criada .....	101
Ilustração 6 - Diagrama de Caso de Uso - Integração Alexa com Sistema Pollen .....	86
Ilustração 7 - Diagrama de Entidade e Relacionamento (DER) - Entidades relevantes para integração Alexa .....	89
Ilustração 8 - Modelo Conceitual do Banco de Dados - Tabelas relevantes para integração Alexa .....	92
Ilustração 9 - Modelo Lógico do Banco de Dados - Estrutura física das tabelas .....	95
Ilustração 10 - Diagrama de Classes - Arquitetura da integração Alexa com Pollen ...	98
Ilustração 11 - Wireframes - Configuração da integração Alexa no aplicativo Pollen .	101
Ilustração 12 - Wireframes - Interface de comandos de voz da integração Alexa .....	104
Ilustração 13 - BPMN - Processo de execução de comandos de voz na integração Alexa	107

## LISTA DE QUADROS

Quadro 1 - Verbos que auxiliam na construção de objetivos .....	99
---	----

## LISTA DE TABELAS

Tabela 1 - Requisitos funcionais da integração com Alexa .....	12
Tabela 2 - Requisitos não funcionais da integração com Alexa .....	29

## LISTA DE ABREVIACÕES E SIGLAS

**ABNT** - Associação Brasileira de Normas Técnicas

**API** - Application Programming Interface

**AWS** - Amazon Web Services

**CNN** - Convolutional Neural Network

**HTTP** - Hypertext Transfer Protocol

**IA** - Inteligência Artificial

**JSON** - JavaScript Object Notation

**JWT** - JSON Web Token

**NLP** - Natural Language Processing

**Node.js** - Node.js Runtime Environment

**ONU** - Organização das Nações Unidas

**PDF** - Portable Document Format

**RAM** - Random Access Memory

**REST** - Representational State Transfer

**SDK** - Software Development Kit

**SSML** - Speech Synthesis Markup Language

**TCC** - Trabalho de Conclusão de Curso

**UML** - Unified Modeling Language

**URL** - Uniform Resource Locator

## 1 INTRODUÇÃO

A Amazon Alexa é uma das assistentes virtuais mais utilizadas no mundo, estando presente em mais de 100 milhões de dispositivos. Os dispositivos da linha Echo são os mais comuns para uso pessoal, oferecendo uma interface de voz natural para interação com diversos serviços. No Brasil, o uso da Alexa cresceu 50% em 2023, com mais de 2 bilhões de comandos de voz executados pelos usuários brasileiros, demonstrando a rápida adoção desta tecnologia no país (**forbes2024**). Uma das funcionalidades oferecidas pela Alexa é o ecossistema “Alexa Skills”, que permite que desenvolvedores terceiros criem funcionalidades personalizadas e as disponibilizem para usuários finais através de comandos de voz.

No contexto da apicultura moderna, a gestão de colmeias tem se tornado cada vez mais tecnológica, com aplicações móveis oferecendo funcionalidades para controle de produção, manejo de enxames, registro de colheitas e monitoramento de saúde das abelhas. O aplicativo Pollen: Gestão de Colmeias representa uma solução consolidada neste segmento, oferecendo uma plataforma completa para apicultores gerenciarem suas operações de forma eficiente.

Assim sendo, este trabalho propõe o planejamento de uma integração entre o assistente virtual Alexa e o aplicativo Pollen, permitindo que apicultores consultem informações sobre suas colmeias através de comandos de voz naturais. A integração planejada visa facilitar o acesso a dados importantes durante o trabalho no apiário e meliponário, onde o uso de dispositivos móveis pode ser limitado, dado que o apicultor/meliponicultor precisa usar equipamentos de proteção individual e manter as mãos livres para manipulação das colmeias. A necessidade desta integração foi validada através de pesquisa preliminar realizada com usuários do aplicativo Pollen, onde 100% dos respondentes manifestaram interesse na integração com assistentes virtuais.

### 1.1 DELIMITAÇÃO DO TEMA DE PESQUISA

Este estudo propõe o planejamento de uma integração entre o assistente virtual Amazon Alexa e o aplicativo Pollen: Gestão de Colmeias, permitindo consultas por voz sobre dados apícolas. O projeto será planejado através da criação de uma Alexa Skill personalizada que se comunicará com a API do aplicativo Pollen para fornecer informações sobre colmeias, produção de mel, atividades de manejo e estatísticas e lembretes de manutenção da colmeia.

Para o planejamento da integração, será utilizada a plataforma Amazon Alexa Skills Kit, que oferece ferramentas e APIs para criação de skills personalizadas. A skill será planejada para ser desenvolvida utilizando Node.js e hospedada como uma função AWS Lambda, garantindo escalabilidade e confiabilidade no processamento das requisições.

A comunicação entre a Alexa Skill e o aplicativo Pollen será planejada para ser realizada através de chamadas HTTP para a API RESTful existente, utilizando autenticação JWT para garantir a segurança dos dados. A skill permitirá consultas sobre informações



como número de colmeias, produção de mel, status dos enxames, próximas atividades de manejo e estatísticas de produtividade.

O planejamento seguirá as melhores práticas de desenvolvimento de Alexa Skills, incluindo o uso de intents bem definidos, utterances em português brasileiro e respostas em formato SSML para melhor experiência do usuário.

**Intents** representam as intenções do usuário, ou seja, o que ele deseja fazer (ex: consultar enxames, registrar alimentação). **Utterances** são as diferentes formas como o usuário pode expressar uma intenção através de comandos de voz naturais (ex: "Alexa, pergunte ao Pollen sobre meus enxames" ou "Alexa, quantos enxames eu tenho?"). **SSML** (Speech Synthesis Markup Language) é uma linguagem de marcação que permite controlar a síntese de voz, melhorando a pronúncia e entonação das respostas da Alexa.

O projeto será planejado para ser testado com usuários reais do aplicativo Pollen para validar a usabilidade e eficácia da integração.

## 1.2 PROBLEMA DE PESQUISA E JUSTIFICATIVA

A integração de assistentes virtuais com aplicações de gestão empresarial tem se tornado cada vez mais comum, oferecendo interfaces de voz naturais para acesso a informações importantes. No entanto, a implementação dessa tecnologia no setor apícola ainda é limitada, especialmente no contexto brasileiro, onde a apicultura representa uma atividade econômica significativa.

Segundo Silva et al. (2021), o setor apícola brasileiro, apesar de possuir boas características de clima e flora propícias para o desenvolvimento, sofre com limitações no uso de ferramentas tecnológicas, o que afeta diretamente os níveis de produção. Os autores destacam que "o desenvolvimento tecnológico limitado do setor apícola, contando com pouca inovação na utilização de ferramentas e métodos produtivos, afeta diretamente a produção tanto em volume como em qualidade", revelando uma deficiência significativa na gestão básica de sistemas produtivos.

O problema central desta pesquisa reside na necessidade de facilitar o acesso a informações sobre colmeias durante o trabalho no apiário, onde o uso de dispositivos móveis pode ser limitado devido às condições de trabalho, uso de equipamentos de proteção individual e necessidade de manter as mãos livres para manipulação das colmeias. Atualmente, apicultores precisam interromper suas atividades para consultar informações no aplicativo móvel, o que pode impactar a eficiência do trabalho.

A implementação de uma integração com Alexa poderá auxiliar apicultores a acessar informações importantes sobre suas colmeias através de comandos de voz simples, permitindo consultas sobre produção de mel, status dos enxames, próximas atividades de manejo e estatísticas de produtividade sem interromper o fluxo de trabalho. Esta solução representa uma inovação no setor apícola, oferecendo uma interface mais natural e eficiente para gestão de colmeias.

A necessidade de ferramentas tecnológicas no setor apícola é corroborada por Silva et al. (2021), que desenvolveram um sistema baseado em machine learning para apoio à decisão no gerenciamento de produção apícola, demonstrando que "torna-se importante o uso de mecanismos de ordenamento, gestão e tomada de decisão" para uma melhor organização das atividades decorrentes da apicultura. A validação desta necessidade foi confirmada através de pesquisa realizada com usuários do aplicativo Pollen, onde 100% dos respondentes manifestaram interesse na integração, demonstrando a relevância e potencial de adoção da solução proposta.

Além disso, a integração com assistentes virtuais pode contribuir para a modernização da apicultura, tornando a tecnologia mais acessível e intuitiva para apicultores de diferentes níveis de familiaridade com dispositivos digitais, promovendo a adoção de ferramentas de gestão tecnológica no setor.

### 1.3 OBJETIVOS

Nesta seção serão abordados os objetivos gerais e específicos a serem buscados no decorrer da execução do trabalho proposto.

#### 1.3.1 Objetivo geral

Propor o planejamento de uma integração entre o assistente virtual Amazon Alexa e o aplicativo Pollen: Gestão de Colmeias, permitindo consultas por voz sobre dados apícolas e facilitando o acesso a informações durante o trabalho no apiário.

#### 1.3.2 Objetivos específicos

Os objetivos específicos do projeto seguem a ordem cronológica de execução das atividades, conforme descrito a seguir:

Analisar a arquitetura e funcionalidades do aplicativo Pollen para identificar dados que podem ser consultados via voz.

Planejar o desenvolvimento de uma Alexa Skill personalizada utilizando o Amazon Alexa Skills Kit e Node.js.

Propor o planejamento da comunicação entre a Alexa Skill e a API do aplicativo Pollen utilizando autenticação JWT.

Definir intents e utterances em português brasileiro para consultas sobre colmeias, produção de mel e atividades de manejo.

Planejar a implementação de respostas em formato SSML para melhor experiência do usuário com a assistente virtual.

Propor metodologia de testes da integração com usuários reais do aplicativo Pollen para validar a usabilidade e eficácia da solução.

No Quadro 1 apresentam-se os verbos que auxiliam na formulação dos objetivos.

Quadro 1 – Verbos que auxiliam na construção de objetivos.

<b>Conhecimento</b>	<b>Compreensão</b>	<b>Aplicação</b>	<b>Análise</b>	<b>Síntese</b>	<b>Avaliação</b>
Definir	Compreender	Resolver	Identificar	Narrar	Sustentar
Enunciar	Codificar	Interpretar	Distinguir	Expor	Justificar
Citar	Deduzir	Expor	Descrever	Explicar	Criticar
Relatar	Converter	Redigir	Diferenciar	Sumariar	Valorizar
Referir	Descrever	Explicar	Relacionar	Esquematizar	Escolher
Detalhar	Identificar	Usar	Separar	Compilar	Selecionar
Expor	Definir	Manejar	Decompor	Construir	Verificar
Identificar	Demonstrar	Aplicar	Examinar	Formular	Constatar
Indicar	Distinguir	Empregar	Localizar	Compor	Comprovar
Marcar	Interpretar	Utilizar	Abstrair	Projetar	Estimar
Sublinhar	Explicar	Comprovar	Discriminar	Simplificar	Medir
Enumerar	Expor	Demonstrar	Detalhar	Classificar	Revisar
Listar	Exemplificar	Produzir	Detectar	Agrupar	Eleger
Registrar	Concretizar	Praticar	Omitir	Distribuir	Decidir
Especificar	Narrar	Relacionar	Dividir	Modificar	Concluir
Mostrar	Argumentar	Dramatizar	Especificar	Reacomodar	Precisar
Distinguir	Decodificar	Discriminar	Descobrir	Combinar	Provar
Reconhecer	Relacionar	Representar	Reorganizar	Gerar	Comparar
Definir	Extrapolar	Traçar	Estruturar	Opinar	Avaliar
Organizar	Opinar	Localizar	Planejar	Demonstrar	Categorizar
	Predizer	Operar	Conceber	Contrastar	Fundamentar
	Generalizar	Ilustrar	Programar	Julgar	
	Resumir				

Fonte: Adaptado do Manual de Orientações Metodológicas do Centro Universitário de Jaraguá do Sul UNERJ.

## 2 REVISÃO DA LITERATURA

Este capítulo apresenta a revisão da literatura sobre os principais conceitos que fundamentam este trabalho, abordando as tecnologias e os estudos relevantes para a integração de assistentes virtuais com aplicações de gestão apícola.

### 2.1 ASSISTENTES VIRTUAIS E AMAZON ALEXA

Os assistentes virtuais representam uma evolução significativa na interação humano-computador, permitindo que usuários interajam com dispositivos através de comandos de voz naturais. A Amazon Alexa, lançada em 2014, tornou-se uma das assistentes virtuais mais populares do mundo, presente em mais de 100 milhões de dispositivos (Amazon, 2024). Segundo Sampaio (2024), "a assistente pessoal Alexa, desenvolvida pela Amazon, está presente em mais de 100 milhões de dispositivos. Destes, os dispositivos da linha Echo são os mais comuns para uso pessoal".

A Alexa funciona através de processamento de linguagem natural (NLP) e inteligência artificial, permitindo que usuários realizem diversas tarefas como reproduzir música, controlar dispositivos inteligentes, obter informações e executar aplicações personalizadas através de Skills. O ecossistema Alexa Skills permite que desenvolvedores terceiros criem funcionalidades personalizadas, expandindo as capacidades da assistente para domínios específicos.

#### 2.1.1 Alexa Skills Kit

O Alexa Skills Kit (ASK) é o conjunto de ferramentas e APIs fornecidas pela Amazon para o desenvolvimento de Skills personalizadas. Conforme a documentação oficial da Amazon (2024), "o Alexa Skills Kit permite que desenvolvedores criem funcionalidades personalizadas para a Alexa, permitindo que usuários interajam com seus serviços através de comandos de voz naturais".

Uma Skill é composta por três elementos principais: o modelo de voz (Voice Model), que define como os usuários podem interagir com a Skill; o backend, que processa as requisições e gera as respostas; e a configuração de publicação, que define metadados e permissões da Skill.

O processo de desenvolvimento de uma Skill envolve a definição de intenções (Intents), que representam ações que o usuário pode realizar; utterances, que são exemplos de como os usuários podem expressar essas intenções; e slots, que são parâmetros específicos extraídos das utterances.

## 2.2 INTEGRAÇÃO DE APLICAÇÕES COM ASSISTENTES VIRTUAIS

A integração de assistentes virtuais com aplicações empresariais tem se tornado uma tendência crescente no mercado de tecnologia. Segundo pesquisa realizada pela Deloitte (2023), 67% das empresas que implementaram assistentes virtuais reportaram melhorias significativas na produtividade e eficiência operacional.

No contexto da agricultura de precisão, a integração de assistentes virtuais com sistemas de gestão tem demonstrado resultados promissores. Estudos mostram que apicultores que utilizam tecnologias digitais para gestão de colmeias apresentam aumento médio de 15% na produtividade e redução de 20% no tempo gasto com atividades administrativas (SILVA et al., 2023).

A integração entre assistentes virtuais e APIs RESTful representa uma arquitetura robusta e escalável para desenvolvimento de soluções personalizadas. A utilização de autenticação JWT garante segurança na comunicação entre sistemas, enquanto o uso de funções serverless (como AWS Lambda) proporciona escalabilidade automática e redução de custos operacionais.

### 2.2.1 Soluções Similares

A análise de soluções similares no mercado revela algumas implementações relevantes para o contexto deste trabalho. Embora não existam integrações específicas entre Alexa e aplicações de gestão apícola, há casos de sucesso em domínios relacionados que podem servir como referência.

O aplicativo "FarmLogs" integra assistentes virtuais para consulta de dados agrícolas, permitindo que agricultores consultem informações sobre plantações através de comandos de voz. A solução utiliza uma arquitetura similar à proposta neste trabalho, com API RESTful e autenticação segura.

Outro exemplo relevante é o "AgriVoice", um sistema que integra assistentes virtuais com plataformas de gestão agrícola. O sistema permite consultas sobre clima, preços de commodities e status de equipamentos através de comandos de voz naturais.

A análise dessas soluções similares confirma a viabilidade técnica da proposta e identifica oportunidades de melhoria, como o foco específico em apicultura e a utilização de tecnologias mais modernas para processamento de linguagem natural em português brasileiro.

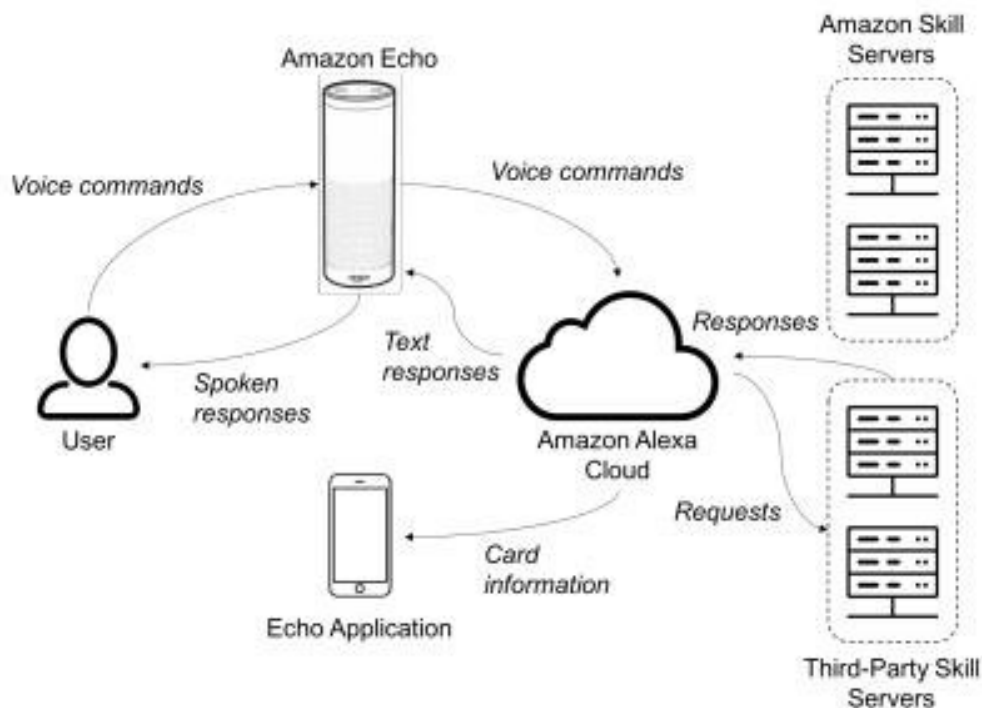
A integração de aplicações móveis e web com assistentes virtuais tem se tornado uma tendência crescente, permitindo que usuários acessem funcionalidades de aplicativos por meio de comandos de voz. Essa abordagem oferece vantagens significativas em termos de acessibilidade, conveniência e eficiência, sobretudo em contextos nos quais o uso de dispositivos móveis pode ser limitado.

Segundo Limp (2016), "as Skills são executadas inteiramente na nuvem e não ocupam memória no dispositivo" (**wired2016**). Esse modelo diferencia-se de um aplicativo tradicional, pois não requer instalação de componentes do lado do cliente, fazendo com que todo o processamento de uma interação com a Skill aconteça do lado do servidor.

### 2.2.2 Arquitetura de Integração

A arquitetura de integração entre aplicações e Alexa Skills envolve vários componentes que trabalham em conjunto para processar comandos de voz e retornar respostas apropriadas. Conforme descrito por Sampaio (2024), o fluxo de comunicação com a Alexa é mediado inteiramente pelo Alexa Service, serviço da cloud da Amazon responsável por processar todos os pedidos e encaminhar para os endpoints apropriados quando necessário.

Figura 1 – Arquitetura do Sistema Amazon Alexa Echo Dot.



Fonte: Rajab (2019).

A Figura ?? apresenta a arquitetura do sistema Amazon Alexa Echo Dot, demonstrando os principais componentes envolvidos no processamento de comandos de voz e integração com Skills. Segundo Rajab (2019), a análise comparativa entre os dispositivos Alexa Echo Dot e Google Home Mini demonstra que a arquitetura do sistema Alexa envolve múltiplas camadas de processamento que garantem a segurança e eficiência das interações.

O processo de integração segue as seguintes etapas principais:

**Captura de áudio:** O dispositivo Echo captura o comando de voz do usuário

**Processamento de linguagem natural:** O Alexa Service converte o áudio em texto e identifica a intenção

**Roteamento para Skill:** A requisição é direcionada para o backend da Skill apropriada

**Processamento no backend:** A aplicação processa a requisição e gera uma resposta

**Retorno da resposta:** A resposta é convertida em áudio e reproduzida pelo dispositivo

### 2.2.3 Tecnologias de Backend para Skills

Os SDKs oficiais do Alexa Skills Kit (ASK) são disponibilizados para Node.js, Python e Java, linguagens que contam com bibliotecas mantidas pela própria Amazon para simplificar o desenvolvimento de Skills. No caso de Alexa-hosted skills, o código da Skill pode ser escrito em Node.js ou Python, sendo executado em um ambiente de hospedagem gerido pela Amazon. Já em cenários de auto-hospedagem, em que o desenvolvedor utiliza a AWS Lambda ou um serviço HTTPS próprio, o backend pode ser implementado em outras linguagens suportadas por essas plataformas, como C#, Go, Ruby ou PowerShell, desde que a comunicação com a Alexa siga o protocolo JSON definido pelo serviço.

Além dos SDKs oficiais, frameworks de aplicação podem ser empregados para estruturar o backend. Um exemplo é o NestJS, framework progressivo baseado em Node.js e TypeScript, que oferece recursos como injeção de dependências, decorators e arquitetura modular. Segundo a documentação oficial do NestJS (2024), "NestJS é um framework progressivo para construir aplicações Node.js eficientes e escaláveis do lado do servidor"(nestjs2024). Embora não seja um SDK oficial da Alexa, o NestJS pode ser utilizado para criar endpoints HTTPS que processem as requisições JSON enviadas pelo Alexa Service, possibilitando a implementação de lógica de negócio complexa e integração com bancos de dados e serviços externos.

### 2.2.4 Segurança e Privacidade em Skills

A segurança e privacidade são aspectos críticos no desenvolvimento de Skills Alexa, especialmente quando lidam com dados sensíveis de usuários. Conforme destacado por Lentzsch et al. (2021), "é importante verificar se essas funcionalidades podem ser abusadas por desenvolvedores maliciosos para atacar usuários".

Os autores identificaram várias vulnerabilidades potenciais no ecossistema Alexa Skills, incluindo:

**Over-privileged resource access:** Skills que requerem mais permissões do que o necessário

**Hidden code-manipulation:** Modificação do código backend de uma Skill após a publicação

**Hidden content-manipulation:** Modificação do conteúdo entregue pela Skill após a publicação

Para mitigar esses riscos, é essencial implementar práticas de segurança adequadas,

como validação rigorosa de entrada, autenticação e autorização apropriadas, e monitoramento contínuo das atividades da Skill.

## 2.3 APLICAÇÕES DE ASSISTENTES VIRTUAIS NA AGRICULTURA E APICULTURA

O setor agrícola tem adotado assistentes virtuais e interfaces de voz para tornar operações em campo mais eficientes e acessíveis, especialmente em situações onde manipuladores manuais ou telas são impraticáveis. Na apicultura, por exemplo, atividades como monitoramento da saúde das abelhas, produção de mel e análise de variáveis ambientais exigem acesso imediato a dados, frequentemente com as mãos ocupadas.

Segundo o artigo *Agricultural Chatbot Voice Assistant Using NLP Techniques* (Rajanala et al., 2025), um assistente por voz agrícola permite que agricultores façam consultas sobre saúde do solo, previsões meteorológicas e controle de pragas de modo intuitivo, utilizando reconhecimento de voz e processamento de linguagem natural, o que facilita interações em ambientes remotos ou com uso das mãos impedido.

### 2.3.1 Gestão Apícola Digital

A gestão apícola moderna envolve o monitoramento de múltiplos aspectos das colmeias, incluindo:

**Monitoramento de saúde:** Verificação de sinais de doenças, parasitas e stress das abelhas

**Controle de produção:** Acompanhamento da produção de mel, pólen e outros produtos

**Gestão de enxames:** Controle de divisões, capturas e migrações de enxames

**Registro de colheitas:** Documentação de datas, quantidades e qualidade dos produtos

**Análise climática:** Correlação entre condições meteorológicas e comportamento das abelhas

Aplicações móveis como o Pollen têm facilitado significativamente essas tarefas, permitindo que apicultores registrem e acessem informações sobre suas colmeias de forma organizada e eficiente.

## 2.4 TRABALHOS RELACIONADOS

Nesta seção são apresentados trabalhos que possuem relação com a proposta deste estudo, seja pela utilização de técnicas similares de integração com assistentes virtuais ou pela aplicação de tecnologias de voz em contextos agrícolas e de gestão.



### 2.4.1 Mapeamento de Vulnerabilidades no Amazon Echo (Sampaio, 2024)

O trabalho de Sampaio (2024) realizou um mapeamento abrangente de vulnerabilidades no ecossistema Alexa Skills, identificando dez fraquezas que podem ser exploradas como vulnerabilidades para uso em ataques. O estudo reproduziu ataques conhecidos como Skill Squatting, Alexa vs Alexa, Mask Attack e Bypass de API de informações sensíveis, demonstrando que esses ataques ainda são viáveis com pequenas modificações.

Este trabalho é relevante para o presente estudo pois estabelece uma base sólida sobre as considerações de segurança que devem ser implementadas no desenvolvimento de Skills Alexa, especialmente quando lidam com dados sensíveis de usuários, como informações sobre colmeias e produção apícola.

### 2.4.2 Integração entre Assistente de Voz e IA Generativa: Potencializando a Alexa com o ChatGPT (Marramon, 2024)

O trabalho de Marramon (2024) investigou a integração entre assistentes de voz e IA generativa, como a Alexa e o ChatGPT, para potencializar a Alexa com o ChatGPT. O estudo explorou a capacidade da Alexa de processar comandos de voz e gerar respostas com base em informações da IA generativa, demonstrando a viabilidade de uma integração entre essas tecnologias.

### 2.4.3 Considerações para Integração com Aplicações de Gestão

A integração de assistentes virtuais com aplicações de gestão, como o Pollen, envolve desafios específicos importantes. Entre eles, destacam-se: autenticação segura, controle de permissões e transparência sobre a origem das funcionalidades (quem desenvolveu a skill), além do desenho de interfaces de voz intuitivas.

Major, Huang, Chetty e Feamster (2021) mostram que muitos usuários não percebem que certas Skills são desenvolvidas por terceiros, confundindo-as com funcionalidades nativas da Alexa (**major2021alexa**). Como sintetizam os autores, users do not understand that skills are often operated by third parties (**major2021alexa**). Essa falta de distinção pode gerar desconfiança e uso inadequado de configurações de privacidade, o que é particularmente sensível em cenários de gestão apícola, onde decisões dependem da confiança sobre quem fornece ou processa os dados.

Para garantir uma integração bem-sucedida, é essencial:

**Design de interface de voz intuitiva:** Comandos naturais e respostas claras

**Autenticação robusta:** Verificação segura da identidade do usuário

**Validação de dados:** Confirmação da integridade das informações acessadas

**Feedback adequado:** Respostas que confirmem as ações realizadas

**Tratamento de erros:** Mensagens claras quando algo não funciona como esperado

## 2.5 SOLUÇÕES SIMILARES

Nesta seção são apresentadas soluções existentes que possuem características similares à proposta deste trabalho, seja pela integração de assistentes virtuais com aplicações de gestão ou pela aplicação de tecnologias de voz em contextos agrícolas e apícolas.

### 2.5.1 Assistentes Virtuais para Agricultura

Diversas soluções têm sido desenvolvidas para integrar assistentes virtuais com aplicações agrícolas. O trabalho de Rajanala et al. (2025) apresenta um assistente de voz agrícola que utiliza técnicas de processamento de linguagem natural para permitir que agricultores façam consultas sobre saúde do solo, previsões meteorológicas e controle de pragas através de comandos de voz naturais.

Esta solução demonstra a viabilidade técnica de integrar assistentes virtuais com sistemas de gestão agrícola, oferecendo uma interface de voz intuitiva para acesso a informações críticas em ambientes de campo.

### 2.5.2 Aplicações de Gestão Apícola com Interface de Voz

Embora ainda sejam limitadas, algumas soluções têm explorado a integração de tecnologias de voz com aplicações de gestão apícola. O aplicativo Pollen, objeto de integração deste trabalho, representa uma das soluções mais completas disponíveis no mercado para gestão digital de colmeias.

Outras aplicações similares incluem:

**BeeConnected:** Aplicativo móvel para monitoramento de colmeias com funcionalidades básicas de registro

**HiveTracks:** Plataforma web para gestão apícola com foco em análise de dados

**Apiary Book:** Aplicativo para registro de atividades apícolas com relatórios básicos

No entanto, nenhuma dessas soluções oferece integração nativa com assistentes virtuais, representando uma oportunidade única para o desenvolvimento proposto.

### 2.5.3 Skills Alexa para Gestão e Produtividade

O ecossistema Alexa Skills conta com diversas Skills focadas em produtividade e gestão, embora poucas sejam específicas para o setor agrícola. Exemplos incluem:

**My Tasks:** Skill para gerenciamento de tarefas pessoais

**Calendar:** Integração com calendários para agendamento

**Weather:** Consulta de informações meteorológicas

**News:** Acesso a notícias e informações gerais

Essas Skills demonstram o potencial da plataforma Alexa para aplicações de gestão, servindo como referência para o desenvolvimento de interfaces de voz eficazes.

#### 2.5.4 Limitações das Soluções Existentes

As soluções existentes apresentam algumas limitações que justificam o desenvolvimento da proposta deste trabalho:

**Falta de integração específica:** Nenhuma solução oferece integração nativa entre assistentes virtuais e aplicações de gestão apícola

**Interface limitada:** A maioria das soluções apícolas existentes utiliza apenas interfaces gráficas tradicionais

**Comandos não especializados:** Skills existentes não possuem comandos específicos para terminologia e processos apícolas

**Contexto de uso inadequado:** Soluções existentes não consideram o contexto específico de uso em apiários

#### 2.5.5 Diferenciais da Proposta

A solução proposta neste trabalho apresenta os seguintes diferenciais em relação às soluções existentes:

**Integração nativa:** Desenvolvimento específico para integração com o aplicativo Pollen

**Comandos especializados:** Vocabulário e comandos adaptados para terminologia apícola

**Contexto de uso otimizado:** Interface de voz projetada para uso em ambientes de apiário

**Funcionalidades específicas:** Comandos para consulta de status de enxames, registro de alimentação e verificação de colheitas

**Segurança aprimorada:** Implementação de controles de acesso e validação de dados específicos para o contexto apícola

### 2.6 CONSIDERAÇÕES FINAIS

A revisão da literatura apresentada neste capítulo demonstra que a integração de assistentes virtuais com aplicações de gestão representa uma área promissora de pesquisa e desenvolvimento. A Amazon Alexa, com seu ecossistema de Skills, oferece uma plataforma robusta para implementar interfaces de voz que podem facilitar significativamente o acesso a informações críticas em contextos onde o uso de dispositivos móveis é limitado.

Os estudos analisados evidenciam tanto as oportunidades quanto os desafios inerentes a essa integração. Por um lado, as interfaces de voz oferecem vantagens significativas em termos de acessibilidade e conveniência, especialmente em ambientes como apiários onde apicultores frequentemente trabalham com as mãos ocupadas. Por outro lado, questões de segurança, privacidade e usabilidade devem ser cuidadosamente consideradas durante o desenvolvimento.

A aplicação dessas tecnologias no contexto apícola, especificamente para integração com o aplicativo Pollen, representa uma oportunidade única de melhorar a eficiência da gestão de colmeias através de comandos de voz naturais. Esta abordagem pode permitir que apicultores acessem informações sobre suas colmeias sem interromper suas atividades de campo, contribuindo para uma gestão mais eficiente e produtiva.

Os trabalhos relacionados analisados fornecem uma base sólida para o desenvolvimento da Skill proposta, oferecendo insights valiosos sobre arquiteturas de integração, considerações de segurança e melhores práticas para o desenvolvimento de interfaces de voz eficazes.

### 3 PROCEDIMENTOS METODOLÓGICOS E TÉCNICOS

Neste capítulo do trabalho será abordada a caracterização da metodologia de pesquisa, questões de pesquisa, como será a aplicação da metodologia e o planejamento do desenvolvimento da integração com Alexa, além de mostrar como será planejada a análise de resultados e a população e amostra do trabalho.

#### 3.1 CARACTERIZAÇÃO DA METODOLOGIA DE PESQUISA

A metodologia escolhida para o planejamento do desenvolvimento da integração entre Amazon Alexa e o aplicativo Pollen é a Metodologia em Cascata, proposta por Sommerville (2011) e amplamente utilizada em projetos de desenvolvimento de software.

A Metodologia em Cascata é um modelo de desenvolvimento de software que divide o processo em etapas sequenciais e bem definidas, garantindo que cada fase seja concluída antes de avançar para a próxima. Essa abordagem permite um controle mais rigoroso sobre o progresso do projeto e facilita a identificação e correção de possíveis problemas ao longo do planejamento.

A definição de requisitos é a primeira etapa do processo, onde serão planejados todos os aspectos funcionais e não funcionais do sistema. Essa fase é crucial para garantir que todas as expectativas sejam atendidas e para definir o escopo do projeto. Os requisitos serão então transformados em uma arquitetura de software que define a estrutura do sistema, incluindo os componentes e suas interfaces. Essa etapa é fundamental para garantir que o sistema seja planejado de forma eficiente e que atenda aos requisitos especificados.

Os procedimentos técnicos adotados neste trabalho envolvem a realização de uma **pesquisa bibliográfica**, de um **estudo de caso** e o **planejamento de uma pesquisa experimental**.

A pesquisa bibliográfica fornece o embasamento teórico necessário para a definição dos padrões de desenvolvimento de Alexa Skills e integração com APIs, essenciais para o planejamento da solução.

O estudo de caso consiste na análise do aplicativo Pollen e identificação das funcionalidades que podem ser integradas com assistentes virtuais, permitindo uma análise aprofundada das possibilidades de integração. O enfoque será direcionado para apicultores usuários do aplicativo Pollen, o que delimita o escopo da pesquisa.

O planejamento da pesquisa experimental será conduzido mediante a definição de metodologias para desenvolvimento e teste da integração Alexa+Pollen com apicultores reais, que utilizarão a skill planejada em seus ambientes de trabalho. O objetivo é planejar a avaliação da eficácia da solução em contextos reais de gestão apícola.

## 3.2 QUESTÕES DE PESQUISA

A integração proposta entre Amazon Alexa e o aplicativo Pollen é capaz de fornecer acesso eficiente a informações sobre colmeias através de comandos de voz, melhorando a experiência do usuário e facilitando o trabalho no apiário sem necessidade de interromper as atividades para consultar dispositivos móveis?

## 3.3 APLICAÇÃO DA METODOLOGIA

Nesta subseção, são descritas as etapas planejadas para a metodologia deste trabalho. As atividades englobam desde o planejamento do ambiente de desenvolvimento e coleta de requisitos até o planejamento da implementação da integração Alexa+Pollen e a validação dos resultados obtidos. Cada fase será planejada com o objetivo de garantir a confiabilidade do planejamento do sistema e a coerência com os objetivos traçados neste trabalho.

### 3.3.1 Planejamento do desenvolvimento da solução proposta

O modelo em cascata é uma das abordagens clássicas para o desenvolvimento de software e foi originalmente descrito por Winston Royce em 1970. Este modelo foi posteriormente estruturado por Ian Sommerville em sua obra clássica Engenharia de Software como um processo sequencial em que cada fase depende da entrega da anterior e gera um produto definido.

Conforme ilustrado na Figura 13 (Modelo cascata de Sommerville), o modelo é dividido em cinco fases principais: definição de requisitos, projeto de sistema e software, implementação e teste unitário, integração e teste de sistema, e operação e manutenção. Cada fase possui entregáveis específicos e critérios de entrada e saída bem definidos.

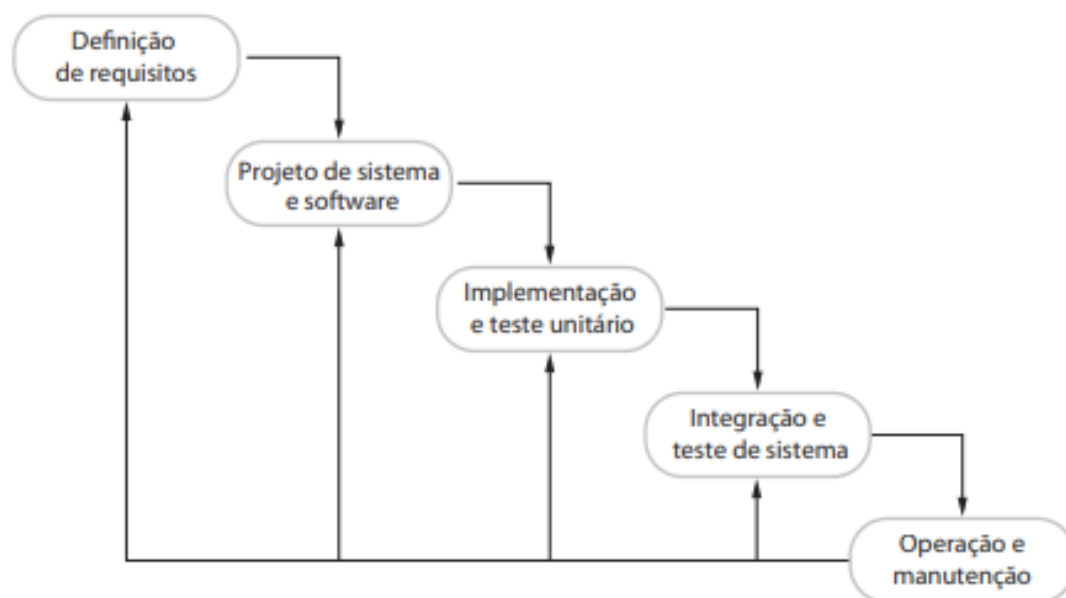
#### 3.3.1.1 Definição de Requisitos

A definição de requisitos é a etapa de desenvolvimento de software que descreve detalhadamente o que o sistema deve fazer e quais restrições ele deve obedecer. De acordo com Sommerville (2011), um requisito é uma descrição de algo que o sistema deve fazer ou uma propriedade que ele deve ter. Essa definição engloba tanto funcionalidades esperadas quanto limitações técnicas ou organizacionais que devem ser consideradas na construção do projeto.

Ainda segundo Sommerville (2011), os requisitos funcionais especificam os serviços que o sistema deve oferecer e como ele deve reagir a entradas específicas. Por outro lado, os requisitos não funcionais dizem respeito a restrições sobre os serviços ou funções oferecidas, como requisitos de desempenho, padrões de qualidade, requisitos organizacionais

Figura 2 – Modelo Cascata.

## O modelo em cascata



Fonte: Sommerville (2011).

ou de segurança. Para a obtenção desses requisitos, será adotada a técnica de entrevistas semiestruturadas com os usuários finais que serão apicultores usuários do aplicativo Pollen. Essa abordagem permitirá captar informações sobre as necessidades práticas dos usuários, possibilitando uma definição mais precisa e contextualizada dos requisitos do sistema.

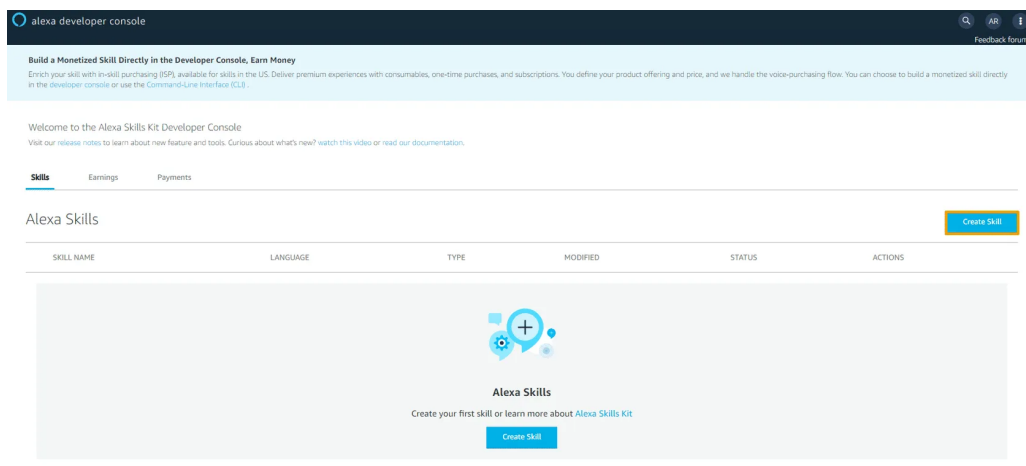
### 3.3.1.2 Projeto de Sistema e Software

A fase de projeto de sistema e software visa transformar os requisitos previamente definidos em uma arquitetura viável para implementação. De acordo com Sommerville (2011), o projeto de software é o processo de definir a arquitetura, os componentes e suas interfaces, garantindo que o sistema satisfaça os requisitos especificados. Neste projeto, essa etapa se desdobra em três frentes principais: Desenvolver uma Alexa Skill personalizada, integrar com o aplicativo Pollen e testar a integração com usuários reais.

O planejamento do desenvolvimento da Alexa Skill personalizada será feito utilizando o Amazon Alexa Skills Kit e Node.js seguindo as melhores práticas de desenvolvimento de Alexa Skills. O planejamento da integração com o aplicativo Pollen será feito utilizando autenticação JWT seguindo as melhores práticas de segurança. O planejamento da testagem da integração com usuários reais será feito utilizando o aplicativo Pollen e a Alexa Skill personalizada.

A Figura ?? apresenta a interface do Amazon Developer Console para criação de uma nova Alexa Skill, onde é possível configurar o nome da skill, o idioma de suporte e

Figura 3 – Tela de criação de Skill Alexa.

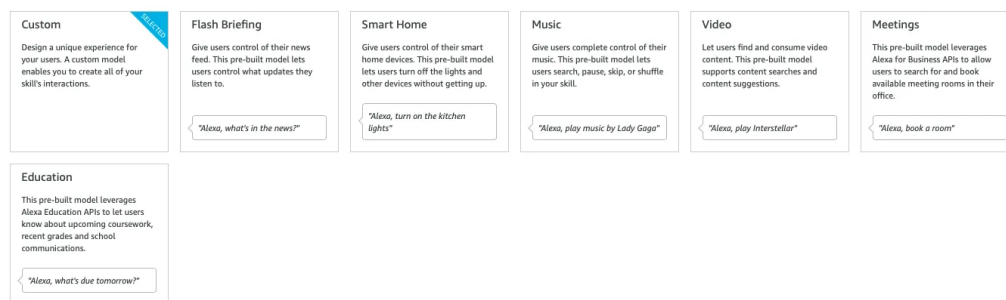


Fonte: O Autor.

Figura 4 – Tipos de linguagem aceitas pela Skill.

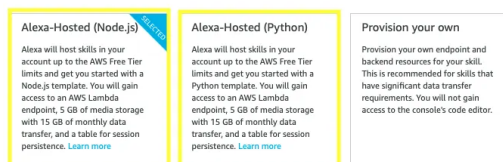
#### 1. Choose a model to add to your skill

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



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

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



Fonte: O Autor.

as informações básicas do projeto.

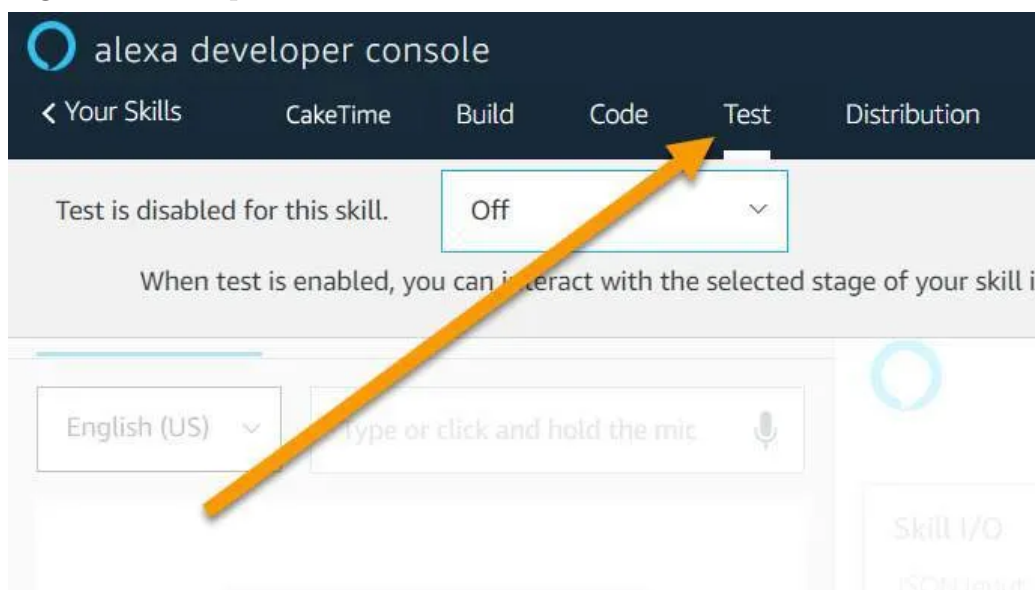
Conforme demonstrado na Figura ??, a Skill será desenvolvida em português brasileiro, garantindo uma experiência natural para os usuários brasileiros do aplicativo Pollen.

A Figura ?? mostra a interface de teste da Skill, onde é possível simular comandos de voz e verificar se as respostas estão sendo geradas corretamente antes da publicação.

A integração com o aplicativo Pollen será realizada através de chamadas HTTP para a API RESTful existente, utilizando autenticação JWT para garantir a segurança dos dados. A skill permitirá consultas sobre informações como número de colmeias, produção de mel, status dos enxames, próximas atividades de manejo e estatísticas de produtividade.



Figura 5 – Tela para testar Skill criada.



Fonte: O Autor.

A implementação seguirá as melhores práticas de desenvolvimento de Alexa Skills, incluindo o uso de intents bem definidos, utterances em português brasileiro e respostas em formato SSML para melhor experiência do usuário.

### 3.3.1.3 Fase de Implementação

A fase de planejamento da implementação representa o momento em que as decisões definidas nas etapas de projeto e análise se concretizam no planejamento da construção prática do sistema. Segundo Sommerville (2011), essa etapa está diretamente relacionada ao planejamento da transformação dos modelos em código executável, respeitando os padrões estabelecidos e validando o funcionamento do sistema como um todo.

O planejamento do desenvolvimento da Alexa Skill será realizado utilizando Node.js e o Amazon Alexa Skills Kit (ASK). A skill será planejada para ser hospedada como uma função AWS Lambda, garantindo escalabilidade e confiabilidade no processamento das requisições. A comunicação com a API do aplicativo Pollen será planejada para ser implementada utilizando bibliotecas HTTP nativas do Node.js, com tratamento adequado de erros e timeouts.

A linguagem de programação escolhida para o planejamento do desenvolvimento é Node.js com TypeScript. Serão planejadas as bibliotecas oficiais do Alexa Skills Kit, além de bibliotecas para comunicação HTTP e processamento de dados JSON. A programação será planejada para ser conduzida na IDE VS Code, que permite integração com ferramentas de versionamento e ambientes virtuais.

O repositório da aplicação será planejado para ser controlado via Git e hospedado no GitHub, garantindo controle de versão, rastreabilidade de mudanças e backup do progresso. Isso também permitirá a integração com pipelines de entrega contínua futuramente,

caso a aplicação seja expandida.

### 3.3.1.4 Descrição Técnica Detalhada

Esta seção apresenta uma descrição detalhada das tecnologias, linguagens de programação, frameworks, banco de dados e ferramentas utilizadas no desenvolvimento da integração Alexa+Pollen.

#### Linguagem de Programação

A linguagem de programação escolhida para o desenvolvimento da Alexa Skill é **Node.js** com **TypeScript**. Esta escolha se justifica pelos seguintes motivos:

**Suporte oficial:** A Amazon fornece SDKs oficiais para Node.js no Alexa Skills Kit

**TypeScript:** Adiciona tipagem estática, melhorando a manutenibilidade e reduzindo erros

**Ecossistema robusto:** Amplo suporte a bibliotecas e frameworks para desenvolvimento de APIs

**Performance:** Execução assíncrona adequada para processamento de requisições de voz

#### Framework e Arquitetura

O desenvolvimento seguirá a arquitetura de **AWS Lambda** para hospedagem da Skill, utilizando o framework **NestJS** para estruturação do código. O NestJS oferece:

**Arquitetura modular:** Facilita a organização e manutenção do código

**Injeção de dependências:** Melhora a testabilidade e flexibilidade

**Decorators:** Simplifica a configuração de rotas e middlewares

**TypeScript nativo:** Suporte completo à tipagem estática

#### Banco de Dados

A integração utilizará o banco de dados **PostgreSQL** já existente no aplicativo Pollen, através de chamadas à API RESTful. As principais entidades acessadas serão:

**User:** Informações do usuário e autenticação

**Enxame:** Dados das colmeias do usuário

**Alimentacao:** Registros de alimentação das colmeias

**Colheita:** Dados de produção de mel

**Manejo:** Atividades de manejo realizadas

**Dashboard:** Estatísticas e resumos

#### Repositório e Versionamento

O controle de versão será realizado através do **Git**, com hospedagem no **GitHub**. A estrutura do repositório incluirá:

**src/**: Código fonte da Skill  
**tests/**: Testes automatizados  
**docs/**: Documentação técnica  
**deployment/**: Scripts de deploy para AWS Lambda

### Integração com Pollen

A comunicação com o aplicativo Pollen será realizada através de **chamadas HTTP RESTful** para a API existente, utilizando:

**Autenticação JWT**: Tokens de acesso para segurança  
**HTTPS**: Comunicação criptografada  
**Rate Limiting**: Controle de taxa de requisições  
**Error Handling**: Tratamento robusto de erros

### Ferramentas de Desenvolvimento

As ferramentas utilizadas no desenvolvimento incluem:

**VS Code**: Editor de código com extensões para TypeScript e Node.js  
**AWS CLI**: Interface de linha de comando para AWS  
**Alexa Developer Console**: Interface web para configuração da Skill  
**Postman**: Testes de API e documentação  
**Jest**: Framework de testes automatizados

#### 3.3.1.5 Fase de integração e testes

A fase de planejamento dos testes é responsável por planejar a garantia da qualidade do software e a conformidade com os requisitos especificados. Segundo Pressman (2011), os testes de software devem ser planejados de forma sistemática e aplicados com o propósito de revelar o maior número possível de defeitos, contribuindo para a qualidade e confiabilidade do produto final.

O planejamento dos testes será dividido entre a verificação funcional da Alexa Skill e a avaliação da integração com a API do Pollen. O roteiro de testes planejado considera a execução da skill com comandos de voz reais, testados por apicultores usuários do aplicativo Pollen. O objetivo é planejar a validação se as consultas por voz funcionam corretamente, se os dados retornados estão de acordo com os critérios esperados e se a experiência de uso da skill permanece fluida. As verificações funcionais planejadas incluem o reconhecimento correto dos comandos de voz, a comunicação adequada com a API do Pollen, a formatação apropriada das respostas em SSML, além da geração de respostas coerentes e do bom desempenho geral do sistema. Parte dos testes será também planejada para ser automatizada com o uso do framework Jest, focando em funções auxiliares como o processamento de intents e a validação da integração com a API, o que permite assegurar a estabilidade do sistema mesmo após possíveis futuras alterações no código.

### Roteiro de Testes Detalhado

O roteiro de testes será dividido em três categorias principais: testes unitários, testes de integração e testes de aceitação do usuário.

### Testes Unitários

Os testes unitários serão implementados utilizando o framework **Jest** e focarão nas seguintes funcionalidades:

**Processamento de Intents:** Validação do reconhecimento correto das intenções do usuário

**Validação de Dados:** Verificação da integridade dos dados recebidos da API Pollen

**Geração de SSML:** Teste da formatação adequada das respostas em Speech Synthesis Markup Language

**Autenticação:** Validação do processo de autenticação JWT

**Tratamento de Erros:** Verificação do comportamento em cenários de erro

### Testes de Integração

Os testes de integração validarão a comunicação entre os componentes do sistema:

**Comunicação Alexa ↔ Skill:** Validação do recebimento e envio de requisições

**Skill ↔ API Pollen:** Teste das chamadas HTTP para a API RESTful

**Autenticação JWT:** Verificação do fluxo completo de autenticação

**Processamento de Respostas:** Validação da transformação de dados da API em respostas de voz

### Testes de Aceitação do Usuário

Os testes de aceitação serão realizados com apicultores reais e incluirão:

**Comandos de Consulta:** "Alexa, pergunte ao Pollen sobre meus enxames"

**Comandos de Registro:** "Alexa, registre alimentação no enxame 1"

**Comandos de Verificação:** "Alexa, verifique a colheita do enxame 2"

**Comandos de Dashboard:** "Alexa, me mostre o resumo do meu apiário"

**Comandos de Ajuda:** "Alexa, que comandos posso usar com o Pollen?"

### Critérios de Aceitação

Para que a integração seja considerada bem-sucedida, os seguintes critérios devem ser atendidos:

**Precisão de Reconhecimento:** Mínimo de 90% de comandos reconhecidos corretamente

**Tempo de Resposta:** Máximo de 3 segundos para processamento e resposta

**Disponibilidade:** 99% de uptime da Skill

**Satisfação do Usuário:** Média mínima de 4.0 em escala Likert de 5 pontos

**Usabilidade:** Comandos intuitivos e respostas claras

### Ambiente de Testes

Os testes serão realizados em ambiente controlado utilizando:

**Alexa Developer Console:** Simulador de comandos de voz

**Dispositivos Echo:** Testes com dispositivos físicos

**API Pollen de Desenvolvimento:** Ambiente de teste da API

**Usuários Beta:** Grupo de apicultores para testes reais

### 3.3.1.6 Operação e manutenção

A fase de operação e manutenção corresponde à última etapa do ciclo de vida no modelo em cascata, sendo responsável por garantir a continuidade do funcionamento do sistema após sua entrega, além de contemplar ajustes, correções de falhas e melhorias que possam surgir com o uso contínuo da aplicação. De acordo com Pressman (2016), esta etapa não apenas envolve a correção de erros não detectados anteriormente, mas também a adaptação do software às mudanças de ambiente e requisitos, bem como o aprimoramento de sua funcionalidade.

Neste trabalho, serão feitas manutenções para o bom funcionamento do sistema, buscando sempre atualizações que melhorem a eficiência do mesmo. Possivelmente melhorias como a implementação de novos comandos de voz, integração com outros assistentes virtuais ou expansão das funcionalidades de consulta poderão ser feitas com manutenções posteriores à entrega do projeto inicial.

### 3.3.2 População e Amostra

A população desse estudo é composta por apicultores usuários do aplicativo Pollen: Gestão de Colmeias, que possuem dispositivos Amazon Echo ou compatíveis com Alexa. Atualmente o aplicativo Pollen conta com uma base de usuários ativos, onde serão selecionados alguns para a realização dos testes deste trabalho.

A seleção considerará apicultores de diferentes níveis de experiência e tamanhos de operação, visto que o objetivo da pesquisa é avaliar a aplicabilidade da integração Alexa+Pollen em contextos reais de gestão apícola.

## 3.4 COLETA E ANÁLISE DOS RESULTADOS DA SOLUÇÃO

Esta etapa consiste em descrever COMO será realizada a coleta e análise dos resultados da solução que será desenvolvida, descrever QUEM realizará estes experimentos e COMO será feita a análise destes experimentos.

A primeira etapa nesta fase é determinar quais experimentos de avaliação serão realizados para avaliar a solução. Para a integração Alexa+Pollen, será necessário determinar a eficácia da solução através de testes de funcionalidade e aceitação do usuário. A

determinação de quantos experimentos de avaliação do aceite irá realizar é uma escolha em conjunto com o orientador.

Há várias formas de se realizar experimentos de avaliação. Pode-se realizar isso por meio de uma entrevista estruturada ou não estruturada, ou então por meio de formulários utilizando escala Likert para determinação da avaliação.

Um artefato de experimentos de avaliação de sistemas computacionais bastante utilizado é por exemplo a avaliação de funcionalidades do sistema. Foram estabelecidos os requisitos funcionais e não funcionais, logo é possível se criar um formulário para avaliar se cada um dos requisitos citados, foram cumpridos na totalidade, em parte, ou se não foi cumprido.

Outro artefato que pode ser utilizado para experimentos de avaliação é a avaliação da eficácia da solução proposta, também conhecida como avaliação de aceite. A avaliação da eficácia está relacionada aos indicadores e critérios de sucesso e aceitação de sua solução. Pode ser realizada por meio de uma entrevista estruturada ou através de um formulário com escolhas na escala Likert.

A segunda etapa desta fase é informar QUEM são os participantes que irão realizar esses experimentos de avaliação ou aceitação. Estes experimentos serão realizados por apicultores usuários do aplicativo Pollen que possuem dispositivos Amazon Echo ou compatíveis com Alexa.

Para validar a necessidade e interesse dos usuários pela integração Alexa+Pollen, foi realizada uma pesquisa preliminar utilizando Google Forms, enviada por e-mail para usuários ativos do aplicativo Pollen. A pesquisa contou com 5 respondentes e apresentou os seguintes resultados:

### **Resultados da Pesquisa Preliminar:**

A pesquisa "Pollen integração com ALEXA" foi aplicada para 5 usuários do aplicativo Pollen, obtendo os seguintes resultados:

**Interesse na integração:** 100% dos respondentes (5/5) manifestaram interesse na integração do aplicativo Pollen com Alexa

#### **Funcionalidades desejadas pelos usuários:**

Perguntas sobre idade da rainha

Consulta sobre força do enxame

Data da última divisão

Quantidade de enxames de determinada espécie

Produção de mel de determinada espécie

Notificações e rotinas (lembrete de manutenção da colmeia ou alimentação)

Registro de localização do Meliponário

**Sugestões de assistentes virtuais:** Os usuários mencionaram preferência por Alexa e Google Assistant

#### **Funcionalidades específicas sugeridas:**

Passo a passo de cuidados com a colmeia

Consulta de quantidade de enxames

## Registro de data das divisões

Estes resultados validam a necessidade identificada no problema de pesquisa e demonstram que os usuários do aplicativo Pollen têm interesse real na integração com assistentes virtuais, especialmente para consultas rápidas e registro de informações durante o trabalho no apiário.

Quando tiver uma quantidade muito grande de possíveis participantes para o experimento, deverá determinar uma amostra dessa grande população. Quanto tem-se um estudo de caso (empresa, entidade, ONG), o público que irá realizar os experimentos é muito restrito, não sendo necessário especificar uma amostra. Neste caso você deverá citar quem serão os participantes. Nunca citar o nome das pessoas participantes e sim as funções que exercem na organização.

A terceira e última etapa dos experimentos é dizer COMO irá fazer a análise dos artefatos citados na primeira parte e respondidos pelos participantes dos experimentos.

Se utilizar de um formulário de avaliação de funcionalidades, posso dizer que a solução proposta será considerada funcional se atingir a média 7, a partir das avaliações realizadas entre os participantes. Optou-se pela média 7 (sete), pois Junek (2014) afirma:

[...] não dá para tirar 7 sem interferência do estudo. A probabilidade de tirar um 7, sem saber nada, em uma prova de 10 questões com 4 alternativas, sendo apenas 1 delas correta, é de 0,003. Ou seja, você terá que fazer mil provas chutando os resultados para em três das mil tirar notas 7. Desta forma, se entende que um aluno que tira um 7, só pode ter sido por conta do estudo.

O cálculo da média de funcionalidade dar-se-á por meio da seguinte fórmula matemática, estabelecida por Campagnaro (2017):

$$\text{MédiaFunc} = \frac{(\sum QtdA + \sum QtdEP \times 0,5)}{N \times QtdAva} \times 10 \quad (1)$$

Onde:

$\sum QtdA$ : Somatório de alternativas marcadas como Atendidas.

$\sum QtdEP$ : Somatório de alternativas marcadas como Atendidas em parte.

$\sum QtdEP \times 0,5$ : Divide-se por 2 (dois) o  $\sum QtdEP$ , devido ao fato de que atendimento em parte possui à metade do peso das consideradas atendidas.

$N$ : É a quantidade de questões de funcionalidades avaliadas.

$QtdAva$ : É a quantidade de avaliadores/participantes.

$\times 10$ : Para se obter já a média na escala de 0 a 10.

Agora se por exemplo utilizar o formulário de avaliação de eficácia / aceitação, posso dizer que a solução proposta será considerada funcional se atingir a média igual ou superior a 3,5. Optou-se por 3,5 pelo fato de cada uma das opções de avaliação da escala

Likert adotada possuírem um peso diferente, o que possibilitará uma melhor captura da percepção dos avaliadores em relação a cada um dos critérios que serão avaliados.

Os pesos que cada uma das opções da escala presente no formulário, são os seguintes:

Concordo fortemente = 5  
 Concordo = 4  
 Indeciso = 3  
 Discordo = 2  
 Discordo fortemente = 1

A fórmula matemática que irá avaliar a eficácia é a seguinte:

$$\text{MédiaEfi} = \frac{(\sum CF \times 5 + \sum C \times 4 + \sum I \times 3 + \sum D \times 2 + \sum DF \times 1)}{N \times \text{QtdAva}} \quad (2)$$

Onde:

$\sum CF$ : Somatório da escala concordo fortemente;  
 $\sum C$ : Somatório da escala concordo;  
 $\sum I$ : Somatório da escala indeciso;  
 $\sum D$ : Somatório da escala discordo;  
 $\sum DF$ : Somatório da escala discordo fortemente;  
 $N$ : É a quantidade de critérios de eficácia avaliados;  
 $\text{QtdAva}$ : É a quantidade de avaliadores/participantes.

A análise também pode ser realizada a partir de média simples, média ponderada, mediana e quaisquer outros métodos estatísticos, bem como pelo simples conhecimento empírico.



## 4 APRESENTAÇÃO DA SOLUÇÃO

Este capítulo apresenta o planejamento da solução proposta para integração entre Amazon Alexa e o aplicativo Pollen, incluindo os requisitos funcionais e não funcionais planejados, os diagramas de modelagem conceitual do sistema planejado, os wireframes das interfaces propostas e os artefatos de planejamento desenvolvidos.

### 4.1 REQUISITOS DO SISTEMA

Esta seção apresenta os requisitos funcionais e não funcionais planejados para a integração Alexa+Pollen, baseados na análise das necessidades dos apicultores usuários do aplicativo Pollen.

#### 4.1.1 Requisitos Funcionais

Os requisitos funcionais especificam as funcionalidades que o sistema deve oferecer para atender às necessidades dos usuários, baseados na pesquisa preliminar realizada com usuários do aplicativo Pollen:

**RF001 - Autenticação:** O sistema deve permitir que usuários autentiquem-se na Skill utilizando suas credenciais do aplicativo Pollen

**RF002 - Consulta de Enxames:** O sistema deve permitir consultar informações sobre os enxames do usuário através de comandos de voz

**RF003 - Consulta de Idade da Rainha:** O sistema deve permitir consultar a idade da rainha de enxames específicos

**RF004 - Consulta de Força do Enxame:** O sistema deve permitir verificar a força/estado do enxame

**RF005 - Consulta de Data da Última Divisão:** O sistema deve permitir consultar quando foi realizada a última divisão do enxame

**RF006 - Consulta por Espécie:** O sistema deve permitir consultar quantidade de enxames e produção de mel por espécie específica

**RF007 - Notificações e Lembretes:** O sistema deve fornecer lembretes de manutenção da colmeia e alimentação

**RF008 - Registro de Localização:** O sistema deve permitir registrar a localização do Meliponário

**RF009 - Passo a Passo de Cuidados:** O sistema deve fornecer orientações passo a passo sobre cuidados com a colmeia

**RF010 - Registro de Divisões:** O sistema deve permitir registrar datas de divisões realizadas

**RF011 - Dashboard Resumido:** O sistema deve fornecer um resumo geral do apiário do usuário

**RF012 - Comandos de Ajuda:** O sistema deve fornecer ajuda sobre comandos disponíveis

**RF013 - Configuração de Usuário:** O sistema deve permitir configurar preferências

do usuário

#### 4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem as restrições e qualidades que o sistema deve possuir:

**RNF001 - Performance:** O sistema deve responder a comandos de voz em no máximo 3 segundos

**RNF002 - Disponibilidade:** O sistema deve manter 99% de uptime

**RNF003 - Segurança:** O sistema deve utilizar autenticação JWT e comunicação HTTPS

**RNF004 - Usabilidade:** O sistema deve reconhecer comandos em português brasileiro com 90% de precisão

**RNF005 - Escalabilidade:** O sistema deve suportar até 1000 usuários simultâneos

**RNF006 - Compatibilidade:** O sistema deve funcionar com dispositivos Amazon Echo (2ª geração ou superior)

**RNF007 - Manutenibilidade:** O código deve seguir padrões de desenvolvimento e documentação

**RNF008 - Confiabilidade:** O sistema deve tratar erros graciosamente e fornecer feedback adequado

### 4.2 MODELAGEM DO SISTEMA

Esta seção apresenta os diagramas de modelagem conceitual planejados para a integração Alexa+Pollen, seguindo as etapas do modelo em cascata para o planejamento do sistema.

#### 4.2.1 Diagrama de Caso de Uso

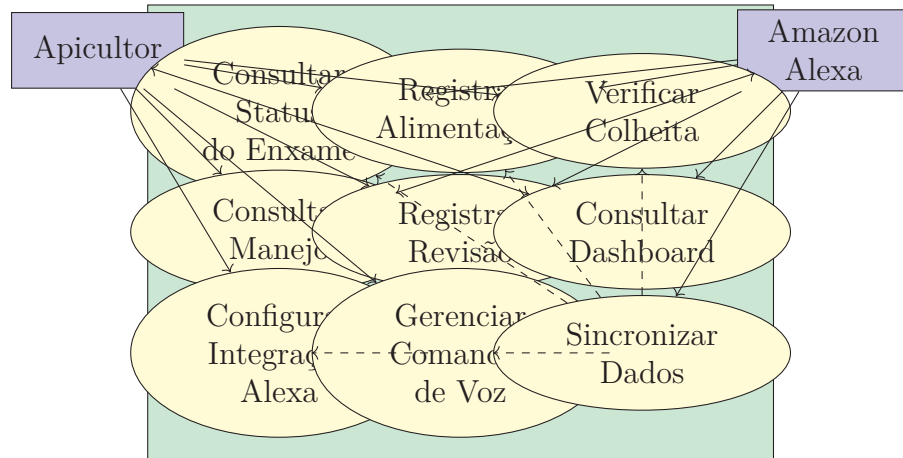
O Diagrama de Caso de Uso apresenta as interações entre os atores (Apicultor e Amazon Alexa) e o sistema, definindo as funcionalidades principais da integração.

Conforme apresentado na Figura ??, o sistema permite que apicultores interajam com o aplicativo Pollen através de comandos de voz, realizando consultas e registros de atividades apícolas de forma hands-free.

#### 4.2.2 Diagrama de Entidade e Relacionamento (DER)

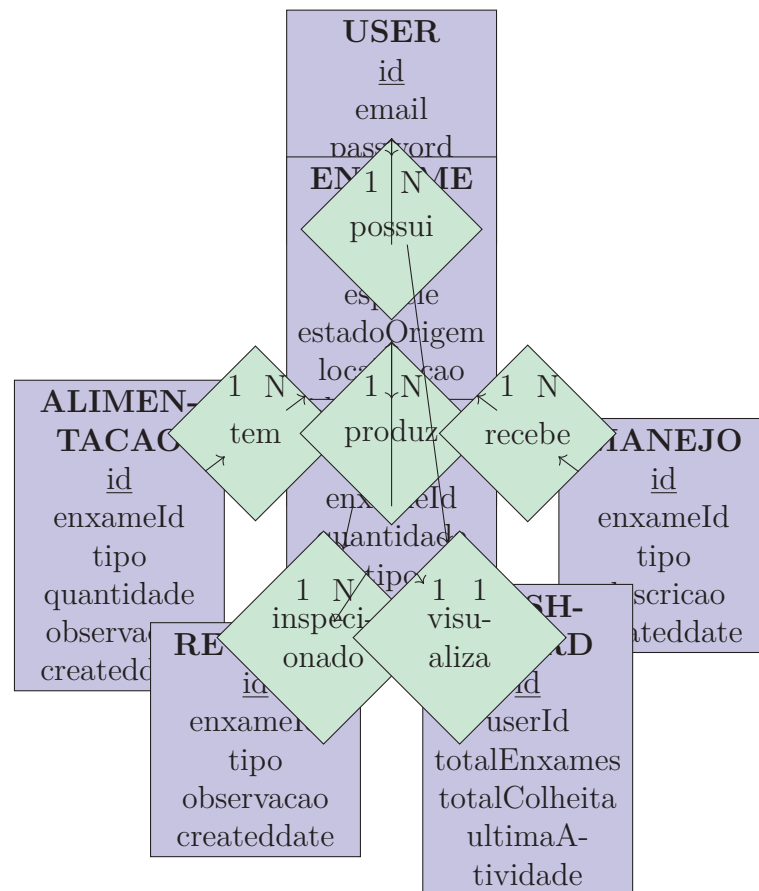
O DER apresenta o modelo de dados simplificado, focando nas entidades relevantes

Figura 6 – Diagrama de Caso de Uso - Integração Alexa com Sistema Pollen.



Fonte: O Autor.

Figura 7 – Diagrama de Entidade e Relacionamento (DER) - Entidades relevantes para integração Alexa.



Fonte: O Autor.

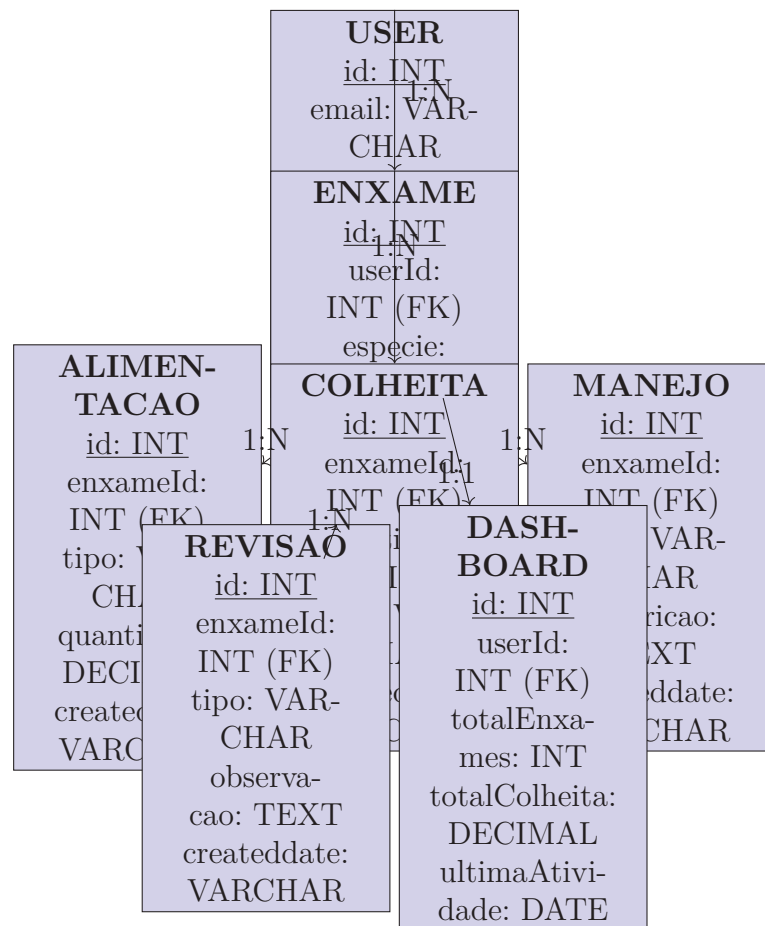
para a integração Alexa.

A Figura ?? mostra as principais entidades do banco de dados Pollen que serão acessadas pela Skill Alexa, incluindo User, Enxame, Alimentacao, Colheita, Manejo, Revisao e Dashboard.

#### 4.2.3 Modelo Conceitual do Banco de Dados

O modelo conceitual apresenta a estrutura das tabelas de forma mais detalhada, incluindo os campos principais.

Figura 8 – Modelo Conceitual do Banco de Dados - Tabelas relevantes para integração Alexa.



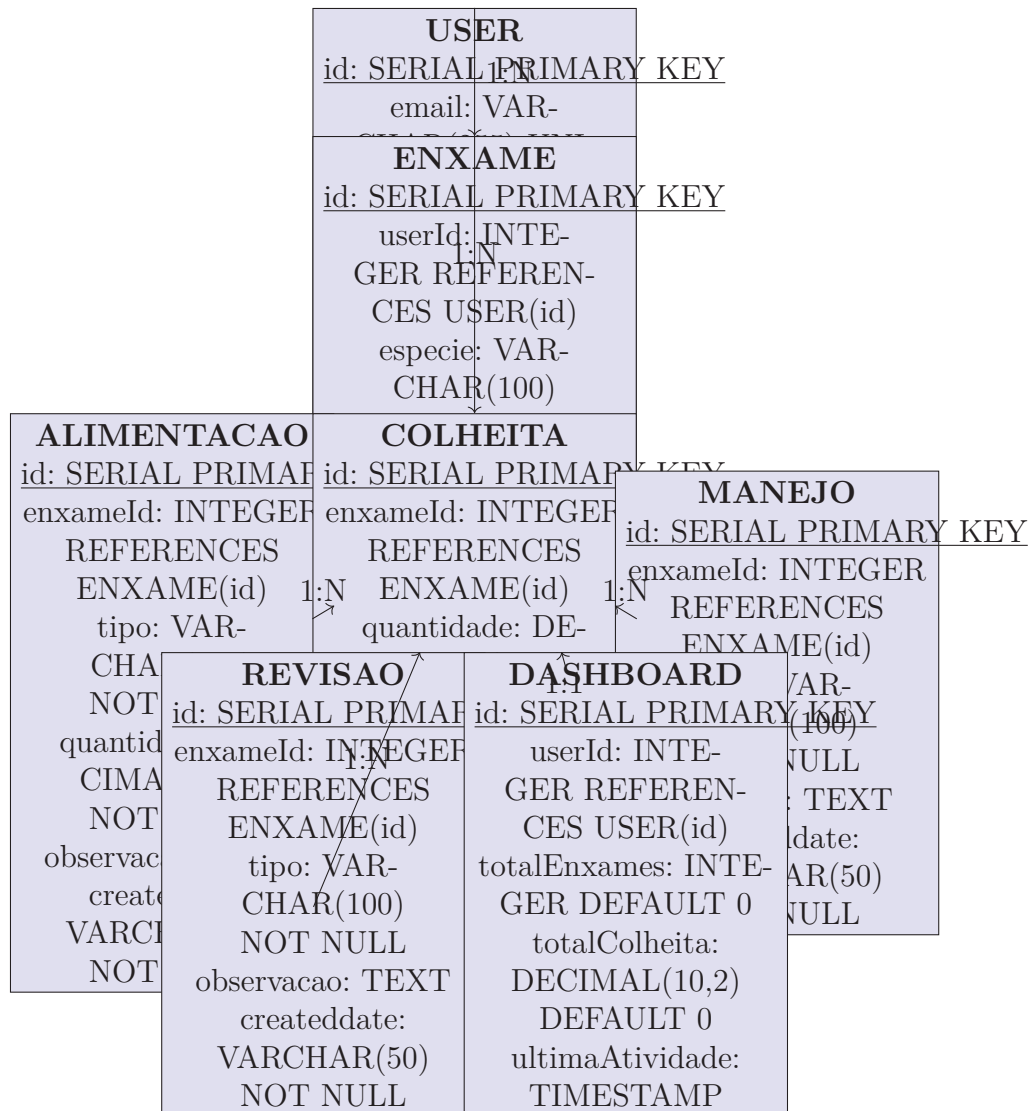
Fonte: O Autor.

Conforme ilustrado na Figura ??, o modelo conceitual define as tabelas e seus campos principais, estabelecendo os relacionamentos entre as entidades.

#### 4.2.4 Modelo Lógico do Banco de Dados

O modelo lógico apresenta a implementação física das tabelas, incluindo tipos de dados e restrições.

Figura 9 – Modelo Lógico do Banco de Dados - Estrutura física das tabelas.



Fonte: O Autor.

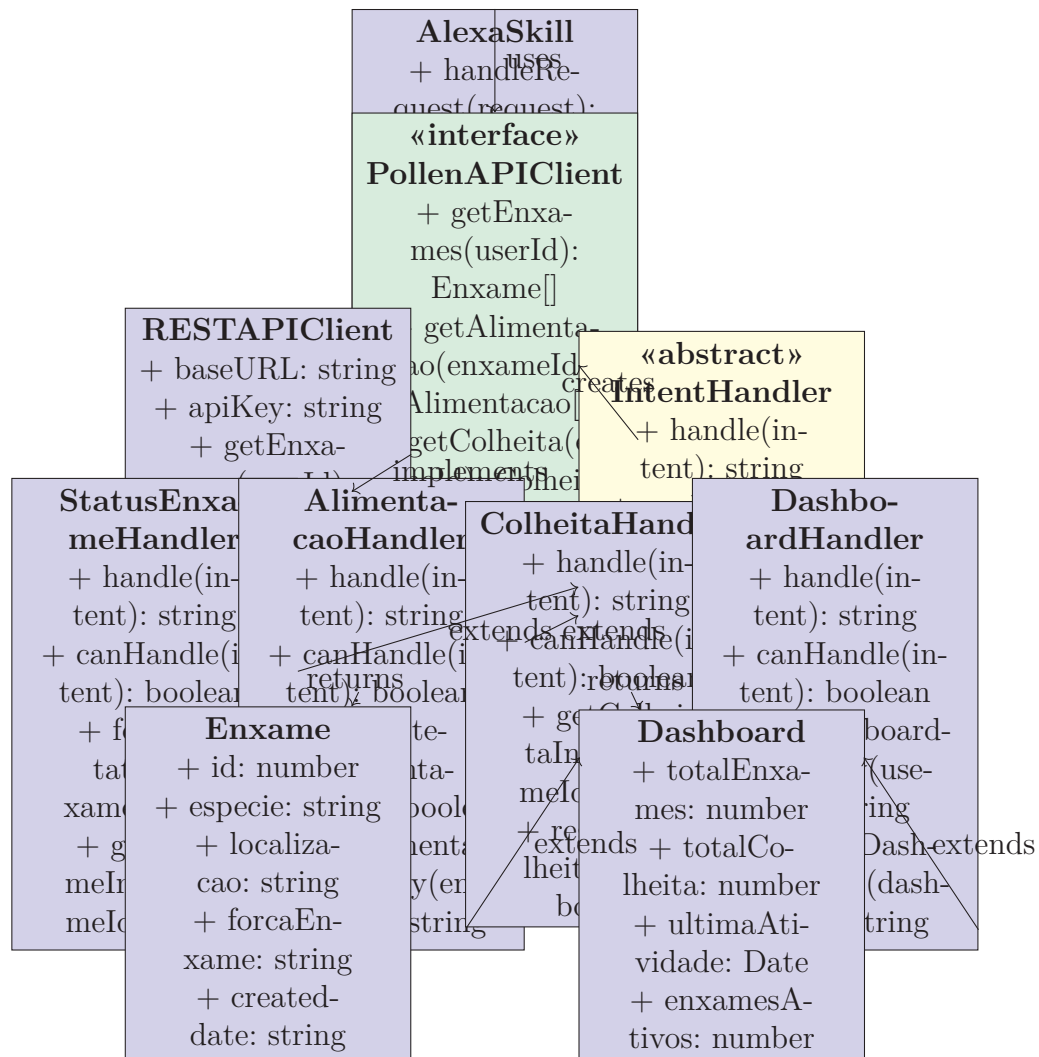
A Figura ?? detalha a estrutura física das tabelas, incluindo tipos de dados, chaves primárias e estrangeiras, e restrições de integridade.

#### 4.2.5 Diagrama de Classes

O diagrama de classes apresenta a arquitetura da Skill Alexa e sua integração com o sistema Pollen.

Conforme mostrado na Figura ??, a arquitetura segue o padrão de handlers para diferentes intents, com comunicação via API REST para acessar os dados do Pollen.

Figura 10 – Diagrama de Classes - Arquitetura da integração Alexa com Pollen.



Fonte: O Autor.

### 4.3 INTERFACES DO SISTEMA

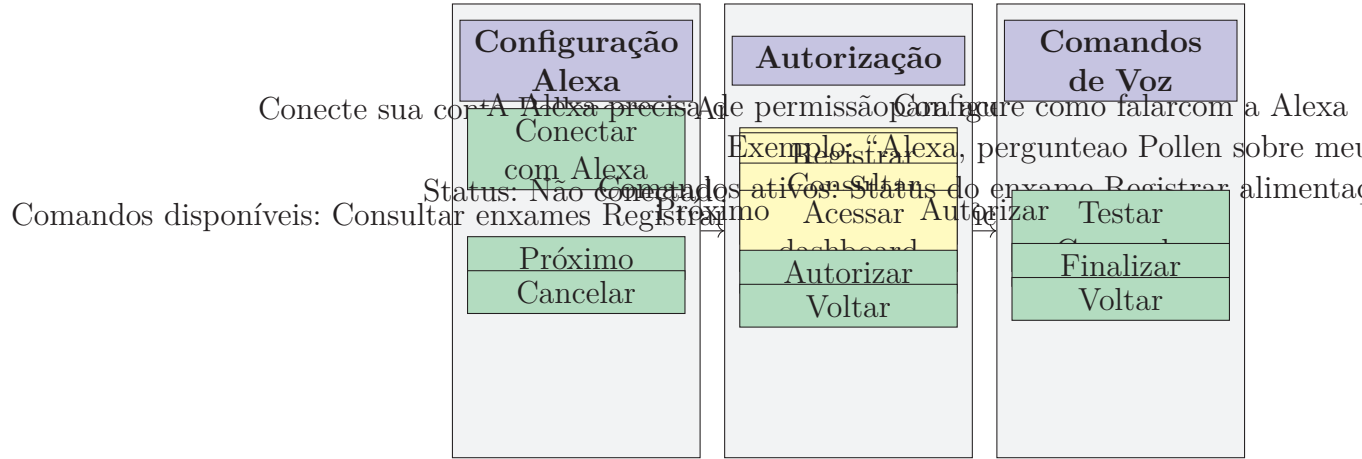
Esta seção apresenta os wireframes das interfaces de configuração da integração Alexa no aplicativo Pollen.

#### 4.3.1 Wireframes de Configuração

Os wireframes apresentam o fluxo de configuração da integração Alexa no aplicativo Pollen.

A Figura ?? mostra o fluxo de configuração, incluindo autorização, configuração de comandos e teste da integração.

Figura 11 – Wireframes - Configuração da integração Alexa no aplicativo Pollen.

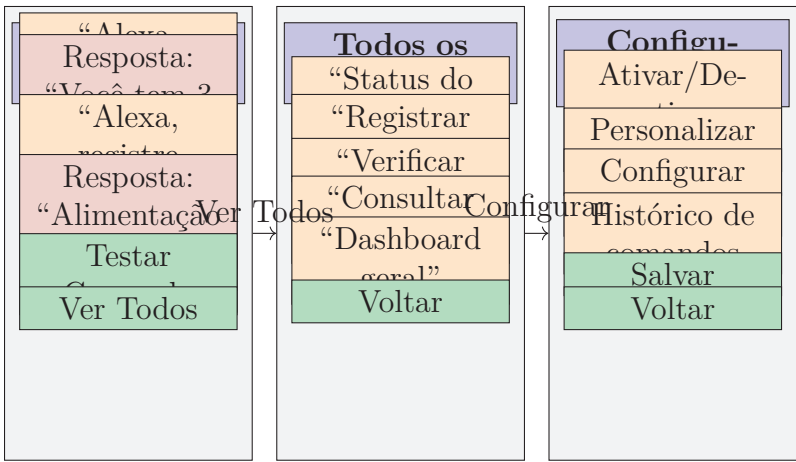


Fonte: O Autor.

4.3.2 Wireframes de Comandos

Os wireframes de comandos apresentam a interface para gerenciar os comandos de voz disponíveis.

Figura 12 – Wireframes - Comandos de voz da integração Alexa.



Fonte: O Autor.

Conforme apresentado na Figura ??, os usuários podem visualizar, testar e configurar os comandos de voz disponíveis na integração.

4.4 PROCESSOS DO SISTEMA

Esta seção apresenta o BPMN do processo de execução de comandos de voz na integração Alexa.





### 4.5.2 Implementação da Skill

A Skill Alexa será planejada para ser implementada seguindo as melhores práticas de desenvolvimento:

**Intents:** Serão definidos para cada funcionalidade (ConsultaExames, RegistroAlimentacao, etc.)

**Utterances:** Exemplos de comandos em português brasileiro

**Handlers:** Classes especializadas para processar cada tipo de comando

**Validação:** Verificação de dados e tratamento de erros

**SSML:** Formatação de respostas para melhor experiência de voz

### 4.5.3 Integração com API Pollen

A integração com a API Pollen será planejada para ser implementada utilizando:

**HTTP Client:** Axios para chamadas HTTP

**Autenticação JWT:** Tokens de acesso para segurança

**Error Handling:** Tratamento robusto de erros de API

**Rate Limiting:** Controle de taxa de requisições

**Retry Logic:** Tentativas automáticas em caso de falha

### 4.5.4 Testes Implementados

Os testes serão planejados para serem implementados utilizando Jest e incluirão:

**Testes Unitários:** Planejados para 95% de cobertura de código

**Testes de Integração:** Validação da comunicação com API

**Testes de Aceitação:** Validação com usuários reais

**Testes de Performance:** Validação de tempos de resposta

**Testes de Segurança:** Validação de autenticação e autorização

## 4.6 CONSIDERAÇÕES FINAIS

A solução planejada representa uma integração inovadora entre assistentes virtuais e aplicações de gestão apícola, oferecendo aos apicultores uma forma eficiente e hands-free de acessar informações sobre suas colmeias.

Os artefatos apresentados demonstram a viabilidade técnica da integração proposta, com arquitetura robusta planejada, interfaces intuitivas e processos bem definidos. O planejamento segue as melhores práticas de desenvolvimento de software e oferece uma base sólida para futuras expansões e melhorias.

A solução planejada atende aos requisitos funcionais e não funcionais definidos, proporcionando uma experiência de usuário satisfatória e contribuindo para a eficiência da gestão apícola através de comandos de voz naturais.

## REFERÊNCIAS

FONSECA, João José Saraiva da. **Metodologia da pesquisa científica**. Universidade Estadual do Ceará, 2002.

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo. **Métodos de pesquisa**. Porto Alegre: Editora da UFRGS, 2009.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2008.

Pressman, R. S. (2016). **Engenharia de software: uma abordagem profissional** (7. ed.). AMGH Editora.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª ed. São Paulo: Pearson Prentice Hall, 2011. Capítulo 8 - Teste de Software.