

AGENDEID

AGENDEID
Ciência da Computação
Versão <1.6>

Histórico da Revisão

Data	Versão	Descrição	Autor
29/04/2025	1.0	Elaboração do documento	Equipe do AGENDEID
3/05/2025	1.1	Base do Chatbot	Equipe do AGENDEID
20/05/2025	1.2	Base solida do Chatbot	Equipe do AGENDEID
14/06/2025	1.3	Criação do Frontend	Equipe do AGENDEID
15/06/2025	1.4	Penúltima versão do Chatbot	Equipe do AGENDEID
20/06/2025	1.5	Versão final prévia do Chatbot	Equipe do AGENDEID
22/06/2025	1.6	Adaptação final da documentação com base na versão real implementada do sistema.	Equipe do AGENDEID

Sumário

1.	Introdução.....	4
1.1	Finalidade.....	4
1.2	Escopo.....	5
1.3	Descrição do Problema.....	5
1.4	Definições, Acrônimos e Abreviações.....	6
1.5	Visão Geral.....	7
1.6	Tabela de Funcionalidades.....	8
2.	Levantamento e Análise de Requisitos.....	8
2.1	Entendimento do Problema.....	8
2.2	Requisitos Funcionais.....	8
2.3	Requisitos Não Funcionais.....	9
3.	Resumo dos Usuários.....	9
4.	Necessidades do Cliente.....	10
5.	Representação Arquitetural.....	11
6.	Metas e Restrições da Arquitetura.....	11
7.	Arquitetura do Sistema e Design de Software.....	12
7.1	Visão Geral da Arquitetura.....	12
7.2	Design de Software.....	12
8.	Visão de Casos de Uso.....	13
9.	Visão Lógica.....	13
9.1	Visão Geral.....	13
9.2	Pacotes de Design Significativos do Ponto de Vista da Arquitetura.....	14
10.	Visão da Implantação.....	15
11.	Visão de Implementação.....	15
12.	Tamanho e Desempenho	16
13.	Qualidade.....	16
13.1	Métricas de Qualidade.....	16

1. Introdução

O **AgendeID** é um chatbot desenvolvido para automatizar e simplificar o processo de agendamento para a emissão da Carteira de Identidade. Através de uma interface conversacional acessível por navegador, o sistema permite que usuários realizem agendamentos diretamente via chatbot sem a necessidade da intervenção humana.

A solução foi projetada para reduzir filas presenciais e simplificar o processo para a emissão da Carteira de Identidade, centralizando a coleta de informações, a validação de dados e a confirmação de agendamentos em um único fluxo automatizado conduzido por um chatbot.

O **AgendeID** opera de forma local e simula um sistema de atendimento digital, com funcionalidades completas para cadastro, login, agendamento, cancelamento, alteração e solicitação de documentos, utilizando um modelo de inteligência artificial treinado para reconhecer as intenções dos usuários.

Esta documentação tem como propósito detalhar a arquitetura do chatbot, os fluxos conversacionais, os requisitos de implantação, além de boas práticas para manutenção e evolução contínua da solução.

1.1 Finalidade

A finalidade do AgendeID é proporcionar uma solução automatizada que facilite o agendamento para a emissão da Carteira de Identidade, oferecendo aos cidadãos um canal de atendimento mais acessível, rápido e eficiente.

Por meio de uma interface conversacional, o chatbot orienta o usuário durante todo o processo de agendamento, desde a coleta das informações pessoais até a seleção do serviço, data e horário.

Além de melhorar a experiência do usuário, o AgendeID contribui para a organização do atendimento e redução da sobrecarga nos postos,

simulando com fidelidade o fluxo de agendamento de documentos de identidade.

1.2 Escopo

Este documento serve como referência para a equipe de desenvolvimento e demais stakeholders do projeto. Ele descreve a estrutura técnica do AgendeID, seus principais fluxos e funcionalidades, com foco na automação do processo de agendamento para emissão da Carteira de Identidade por meio de um chatbot inteligente.

1.3 Descrição do Problema

Problema	<ul style="list-style-type: none"> • Agendamento presencial confuso • Falta de canal digital • Longas filas e espera excessiva
Afeta	<ul style="list-style-type: none"> • Usuários que precisam emitir a Carteira de Identidade (CI). • Servidores responsáveis pelo atendimento
Impacto	<ul style="list-style-type: none"> • Atendimento lento e ineficiente. • Sobrecarga nos postos • Frustração e desistência por parte dos usuários
Solução	<ul style="list-style-type: none"> • Agendamento Digital • Atendimento automatizado, rápido e acessível

1.4 Definições, Acrônimos e Abreviações

MTV – Padrão arquitetural adotado no sistema, separando as responsabilidades em três camadas: modelo, template e visão.

CI – Carteira de Identidade. Documento de identificação pessoal emitido por órgãos públicos.

LGPD – Lei Geral de Proteção de Dados. Legislação brasileira que regulamenta o uso e tratamento de dados pessoais.

Chatbot– Interface automatizada de conversação que interage com o usuário em linguagem natural, facilitando o uso do sistema.

TensorFlow – Biblioteca de aprendizado de máquina usada no sistema para prever demanda e otimizar agendamentos.

NumPy – Biblioteca Python utilizada para processamento e análise eficiente de dados numéricos históricos e operacionais.

Nltk– Biblioteca Python utilizada para o processamento e análise de linguagem natural, oferecendo ferramentas para tokenização, classificação, análise sintática, stemming, lematização e outras tarefas relacionadas ao entendimento e manipulação de textos em linguagem humana.

Pickle – Biblioteca Python utilizada para a serialização e desserialização de objetos, permitindo o armazenamento e a recuperação de estruturas de dados Python em formato binário para posterior reutilização.

SQLite – Sistema de banco de dados utilizado para armazenamento das informações do sistema.

Frontend – Camada visual do sistema, acessada por meio de navegador ou aplicativo móvel, responsável pela interação com o usuário.

Backend – Camada responsável pelo processamento das regras de negócio, integração com banco de dados e comunicação com APIs externas.

Stakeholder – Qualquer parte interessada no projeto, incluindo clientes, atendentes, gestores e desenvolvedores.

1.5 Visão Geral

Neste documento, são apresentadas diferentes visões arquiteturais que descrevem o funcionamento do AgendeID em seus principais processos, desde a interação com o usuário via chatbot até o gerenciamento interno dos dados de agendamento.

Serão abordados aspectos como a implantação local da solução, os componentes envolvidos na implementação, além das restrições e requisitos relacionados a desempenho, segurança, usabilidade e qualidade do serviço.

A proposta é fornecer uma visão clara e detalhada de como o chatbot opera, suas dependências técnicas e os cuidados adotados para garantir a entrega de uma solução funcional, acessível e confiável no contexto simulado de atendimento digital.

1.6 Tabela de Funcionalidades

Funcionalidades
<ul style="list-style-type: none">• Agendamento de serviços via chatbot• Atendimento automatizado com inteligência artificial• Cadastro e login de usuários (clientes e atendentes)• Consulta, alteração e cancelamento de agendamentos• Visualização da agenda e confirmação de presença• Geração de relatórios internos sobre os atendimentos realizados

2. Levantamento e Análise de Requisitos

2.1 Entendimento do Problema

- O processo atual para agendar a emissão da Carteira de Identidade é confuso e pouco acessível.
- Muitos cidadãos não sabem quais documentos levar ou como marcar horário.
- Atendentes enfrentam filas e desorganização por falta de controle dos agendamentos.

- Falta uma solução simples, online e automatizada que facilite esse processo para todos.

2.2 Requisitos Funcionais:

- Cadastro, login e autenticação de clientes e atendentes
- Agendamento, consulta, alteração e cancelamento de atendimentos
- Atendimento automatizado via *chatbot*
- Visualização da agenda e confirmação de presença pelos atendentes
- Geração de relatórios internos sobre os agendamentos realizados

2.3 Requisitos Não Funcionais:

- Autenticação básica para garantir segurança dos dados
- Interface simples, responsiva e acessível via web e mobile
- Suporte a múltiplos usuários acessando o sistema simultaneamente
- Armazenamento local de dados utilizando SQLite
- Backend desenvolvido com Python (Flask) seguindo o padrão MTV
- Frontend implementado com HTML, CSS, JavaScript e
- Uso de NumPy e TensorFlow no processamento de linguagem natural do chatbot
- Tratamento básico de dados pessoais conforme princípios da LGPD

3. Resumo dos Usuários

Nome	Responsabilidade	Descrição
Cliente	Agendar atendimento para emissão da Carteira de Identidade (CI).	Acessa o sistema via navegador, interage com o chatbot para fornecer seus dados, escolhe o serviço, data e horário do atendimento, e recebe um protocolo de confirmação.

Atendente	Gerenciar os agendamentos realizados no sistema.	Acessa o sistema como funcionário autenticado, visualiza a agenda do dia, confirma presença dos clientes e gera relatórios. Toda a interação também é conduzida via chatbot.
-----------	--	--

4. Necessidades do Cliente

Identificador	Necessidade	Prioridade	Situação Atual
Cliente (Caso 1)	Agendar atendimento para emissão da Carteira de Identidade (CI).	Alta	Ir pessoalmente a um posto ou tentar agendar por telefone ou site com processos pouco intuitivos.
Cliente (Caso 2)	Obter informações sobre documentos necessários e horários disponíveis.	Alta	Buscar por informações em sites desatualizados, cartazes físicos ou por atendimento telefônico.
Cliente (Caso 3)	Receber confirmação do agendamento com protocolo claro.	Média	Depender de confirmação manual ou anotações feitas durante visita ao posto de atendimento.

Atendente (Caso 1)	Organizar a agenda de atendimentos de forma digital.	Alta	Controlar agendamentos por planilhas ou de forma manual.
Atendente (Caso 2)	Reduzir o volume de atendimento desorganizado e fora de horário agendado.	Alta	Lidar com demanda espontânea sem controle prévio, gerando filas e sobrecarga no atendimento.
Atendente (Caso 3)	Confirmar e consultar agendamentos com facilidade.	Média	Utilizar sistemas internos desconectados do canal de solicitação do cidadão.

5.Representação Arquitetural

- **Visão de Casos de Uso:**

Apresenta as principais interações do sistema com os usuários (clientes e atendentes), como cadastro, login, agendamento, consulta, alteração, cancelamento, interação com chatbot e geração de relatórios. Todas as interações ocorrem por meio da interface conversacional do chatbot.

- **Visão Lógica:**

O sistema adota uma arquitetura baseada no padrão **MTV (Model-Template-View)** de forma simplificada, utilizando Python com Flask. A lógica de negócio e acesso a dados estão integradas nas views, os templates renderizam a interface (HTML/CSS/JS) e o modelo do chatbot classifica as intenções com o uso de TensorFlow e NumPy.

- **Visão de Processos:**

O fluxo se inicia com a interação do usuário com o chatbot. A cada mensagem, o sistema classifica a intenção, processa a etapa correspondente (ex: agendar, consultar, cancelar) e responde dinamicamente. O banco SQLite armazena os dados de usuários, agendamentos e solicitações.

- **Visão de Implantação:**

O sistema foi desenvolvido para rodar em ambiente local (localhost), acessado por navegador, com suporte básico para dispositivos móveis. A implantação atual não visa uso em produção com múltiplos acessos simultâneos em grande escala, mas sim testes e demonstração funcional.

- **Visão de Implementação:**

Todos os módulos foram desenvolvidos em Python com Flask, utilizando SQLite como

banco de dados local. O chatbot é treinado com TensorFlow, NumPy e NLTK para interpretar comandos dos usuários. A interface é construída com HTML, CSS e JavaScript (vanilla), com comunicação assíncrona via Fetch API.

6. Metas e Restrições da Arquitetura

Metas:

- Entrega de um sistema funcional de agendamento com chatbot integrado
- Simulação completa de um atendimento automatizado
- Acessibilidade por navegador e interface responsiva
- Facilidade de manutenção e organização do código

Restrições:

- Armazenamento local com SQLite
- Utilização de NLTK, NumPy e TensorFlow no back-end (para o modelo de chatbot)
- Interface construída com HTML, CSS, JavaScript (vanilla) e Bootstrap
- Comunicação assíncrona utilizando Fetch API
- Operação em ambiente local (sem integração externa com APIs públicas)
- Modelo de rede neural voltado exclusivamente para classificação de intenções no chatbot

7. Arquitetura do Sistema e Design de Software

7.1 Visão Geral da Arquitetura

Padrão:

- **MTV** – Separação funcional entre dados (Model), apresentação (Template) e controle (View), aplicada de forma simplificada com Flask.

Tecnologias:

- **Backend:** Python com Flask, utilizando a biblioteca NLTK para processamento de linguagem natural no chatbot.
- **Frontend:** HTML, CSS, JavaScript (vanilla), com comunicação assíncrona via Fetch API.
- Banco de Dados: SQLite
- Bibliotecas:

- **NumPy:** Utilizada no pré-processamento de dados de entrada do chatbot (vetorização de palavras).

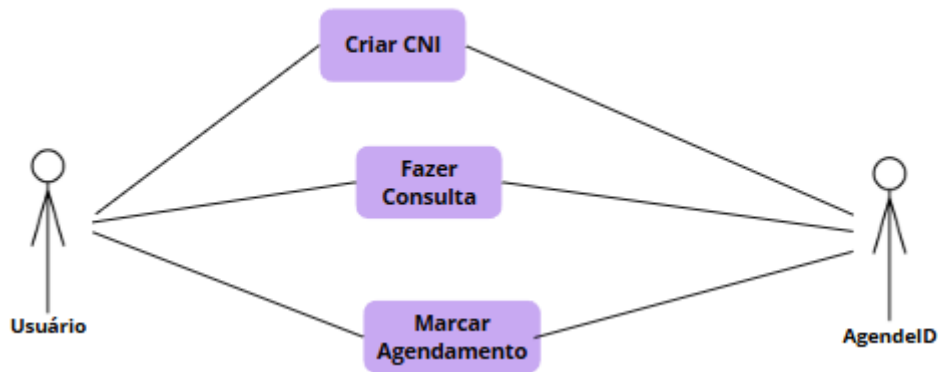
- **TensorFlow:** Utilizada para executar o modelo de rede neural responsável pela classificação de intenções no chatbot.

7.2 Design de Software (Diagramas)

- **UML de Classes:** Descreve os principais dados manipulados no sistema, como usuários e agendamentos.
- **Diagrama de Sequência:** Representa o fluxo de mensagens durante o processo de agendamento via chatbot.
- **Diagrama de Componentes:** Conceitua os módulos lógicos do sistema: “Cadastro”, “Agendamento”, “Relatórios” e “Autenticação”.
- **Modelos Conceitual e Lógico do Banco de Dados:** Estrutura o relacionamento entre as tabelas implementadas em SQLite.

8. Visão de Casos de Uso

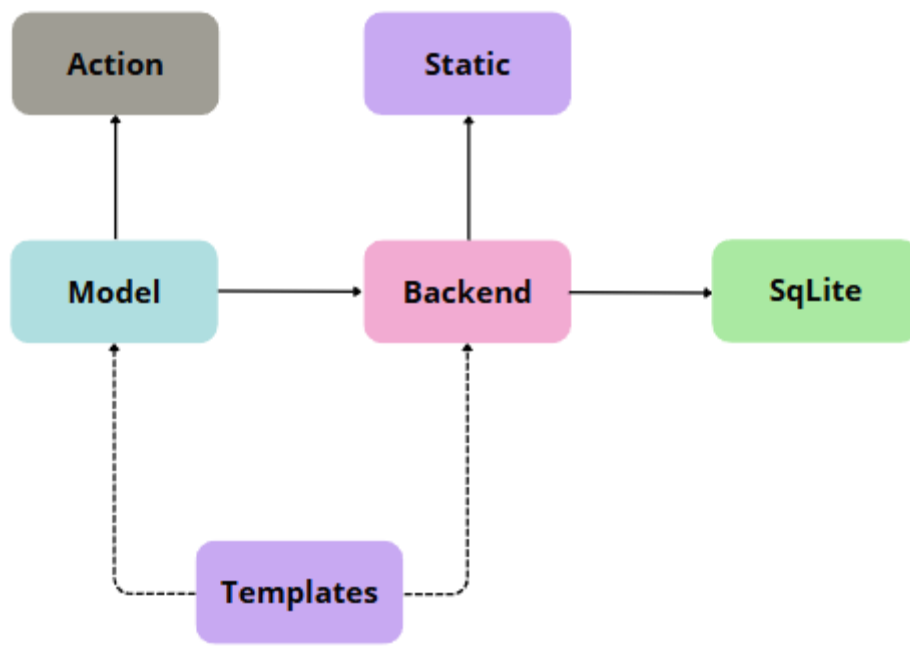
Os casos de uso do sistema **AgendeID** serão listados abaixo:



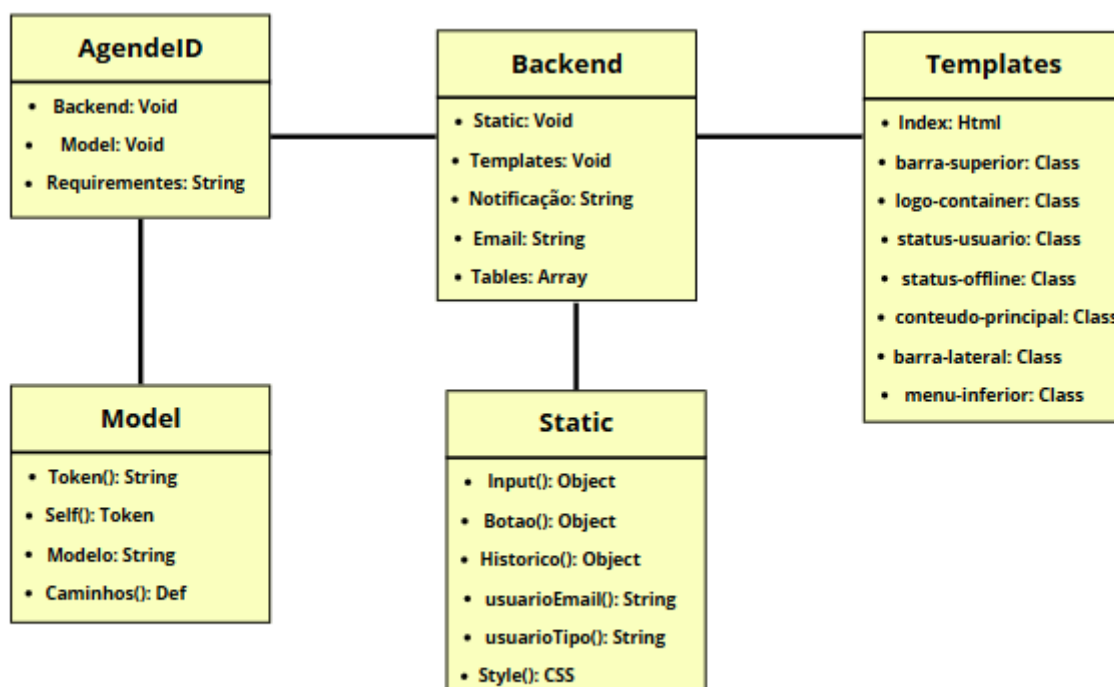
9. Visão Lógica

9.1 Visão Geral

A visão lógica define a estrutura interna da aplicação, organizada com base no padrão MTV (Model-Template-View), utilizando TensorFlow. O sistema é modular, com componentes independentes responsáveis por funcionalidades específicas, como autenticação, agendamento, chatbot e geração de relatórios.



9.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura



10. Visão de Implantação

O sistema AgendeID foi desenvolvido para ser executado localmente, utilizando a stack Python com NLTK, TensorFlow e SQLite. A aplicação é acessada por meio de navegadores em ambiente de testes ou demonstração, com suporte básico à responsividade. A infraestrutura atual não visa produção com múltiplos acessos simultâneos, mas sim a validação completa de todos os fluxos de atendimento por meio de um chatbot funcional. A comunicação entre cliente e servidor ocorre via requisições HTTP assíncronas, utilizando Fetch API para troca de mensagens com o backend.

11. Visão de Implementação

O sistema AgendeID foi implementado de forma modular, com arquivos distintos para lidar com as funcionalidades principais: autenticação, cadastro, agendamento, cancelamento, solicitação de documento, confirmação de presença e geração de relatórios. Todo o processamento é feito no backend, desenvolvido com **Python (Flask)**.

O chatbot utiliza **NLTK**, **NumPy** e **TensorFlow** para o processamento de linguagem natural e classificação de intenções com base em um modelo pré-treinado. A entrada do usuário é interpretada, processada e respondida de forma dinâmica com base no estado da conversa.

O projeto adota o padrão **MTV (Model–Template–View)** de forma simplificada:

- **Model:** manipulação dos dados e conexão com o banco **SQLite**.
- **Template:** interface construída com **HTML**, **CSS** e **JavaScript**.
- **View:** rotas do Flask que coordenam as requisições e respostas do sistema.

A comunicação entre frontend e backend é feita de forma assíncrona com **Fetch API**. O controle de versão é realizado com **Git**, porém **não há pipeline automatizado (CI/CD)** nesta versão. O sistema é voltado para **execução local e testes funcionais**, com estrutura preparada para futura expansão.

12. Tamanho e Desempenho

A aplicação deverá funcionar por meio de um site próprio na versão web, voltado para uso dos atendentes e cidadãos. Também estará disponível na versão mobile, destinada ao uso dos cidadãos para agendamento e consulta. O sistema deve suportar múltiplos acessos simultâneos, sendo o ideal para operações simultâneas de escala moderada.

13. Qualidade

O padrão arquitetural adotado foi pensado para garantir:

- Organização do código-fonte
- Facilidade de manutenção e testes
- Portabilidade da aplicação
- Separação clara de responsabilidades
- Funcionalidade completa dentro de um único ambiente integrado

13.1 Métricas de Qualidade

1. Cobertura de Testes

Foram realizados testes manuais completos nos fluxos de cadastro, login, agendamento, alteração, cancelamento, CIN e relatório. O sistema se comporta de forma estável e previsível.

2. Tempo de Resposta (Percepção do Usuário):

As respostas do chatbot e do sistema ocorrem em poucos segundos durante o uso local.

3. Tratamento de Erros:

O sistema informa erros com mensagens claras e seguras quando detecta entradas inválidas ou ações fora de fluxo.

4. Feedback dos Usuários

Meta: Obter uma média de 4,0/5 nas pesquisas trimestrais de feedback.

5. Satisfação do Usuário:

A navegação guiada pelo chatbot facilita a experiência e reduz falhas por parte do usuário.

6. Escalabilidade:

O sistema é ideal para uso local e simulação. Suporta múltiplos acessos simultâneos em escala moderada.