

Documento de Requisitos IA do Estudante

Ficha Técnica

Equipe Responsável pela Elaboração

Breno Leonardo M.. Braga - Responsável pela elaboração do Documento de Requisitos

Evelyn Santos - Apoio na transcrição da entrevista e estruturação do roteiro

Gabriel Luna - Entrevistador e responsável pelo relato das técnicas

Fernando Quemener - Transcrição da entrevista e apoio na elaboração do documento

Público Alvo

Este manual destina-se a desenvolvedores, alunos universitários, professores e demais envolvidos no projeto do chatbot para o SIGAA.

Sumário

INTRODUÇÃO	P5
Visão geral deste documento	P5
Convenções, termos e abreviações	P5
1. Identificação dos Requisitos	P5
2. Prioridades dos Requisitos	P5
CAPÍTULO 1 - DESCRIÇÃO GERAL DO SISTEMA	C1 . P7
Abrangência e sistemas relacionados	C1 . P7
Descrição dos usuários	C1 . P7
1. Aluno	C1 . P7
2. Professor	C1 . P7
CAPÍTULO 2 - REQUISITOS FUNCIONAIS (CASOS DE USO)	C2 . P8
Consultas Acadêmicas	C2 . P8
[RF001] Consulta de matrícula	C2 . P8
[RF002] Consulta de notas	C2 . P8
[RF003] Emissão de histórico	C2 . P8
[RF004] Informações sobre RU	C2 . P8
Apoio ao Conteúdo de Aula	C2 . P9
[RF005] Acesso ao conteúdo de aula perdida	C2 . P9
[RF006] Trilha de estudo para prova	C2 . P9
CAPÍTULO 3 - REQUISITOS NÃO FUNCIONAIS	C3 . P2
Usabilidade	C3 . P10
[NF001] Linguagem acessível e amigável	C3 . P10
Confiabilidade	C3 . P10
[NF002] Coerência da informação fornecida	C3 . P10
[NF003] Disponibilidade do Serviço	C3 . P10
[NF004] Recuperação de Falhas	C3 . P11
Desempenho	C3 . P11
[NF005] Tempo de resposta rápido	C3 . P11
[NF006] Suporte a Usuários Concomitantes	C3 . P11
[NF007] Otimização do Consumo de Recursos	C3 . P11
Segurança	C3 . P12
[NF008] Gerenciamento de Sessão	C3 . P12
Distribuição	C3 . P12

[NF009] Multiplataforma	C3 . P2
Padrões	C3 . P13
[NF010] Aderência a padrões de acessibilidade	C3 . P13
Hardware e software	C3 . P13
[NF011] Ambiente de Produção Contratado	C3 . P13

Introdução

Este documento tem como objetivo especificar os requisitos para o desenvolvimento de um chatbot destinado a auxiliar alunos universitários no uso da plataforma SIGAA. O sistema, denominado "IA do Estudante", visa responder dúvidas frequentes relacionadas a matrícula, histórico, notas, frequência, conteúdos de aula, cadastro no R.U. e outros serviços acadêmicos.

O documento fornece aos desenvolvedores todas as informações necessárias para o projeto, implementação, testes e homologação do sistema, servindo como referência durante todo o ciclo de desenvolvimento.

Visão geral deste documento

Este documento foi elaborado com base nas entrevistas realizadas com usuários do SIGAA e descreve os requisitos do sistema chatbot "IA do Estudante". As informações foram organizadas da seguinte forma:

- **Seção 2** – Descrição geral do sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 3** – Requisitos funcionais (casos de uso): especifica todos os requisitos funcionais do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
- **Seção 4** – Requisitos não funcionais: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.

Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1. Identificação dos Requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguido do identificador do requisito, de acordo com o esquema abaixo:

[nome da subseção.identificador do requisito]

Por exemplo, o requisito [Recuperação de dados.RF016] está descrito em uma subseção chamada "Recuperação de dados", em um bloco identificado pelo número [RF016]. Já o requisito não funcional [Confiabilidade.NF008] está descrito na seção de requisitos não funcionais de Confiabilidade, em um bloco identificado por [NF008].

2. Prioridades dos Requisitos

Para estabelecer a prioridade dos requisitos foram adotadas as denominações "essencial", "importante" e "desejável".

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que

podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

3. Abreviações e Siglas

- RF – Requisito Funcional
- NF – Requisito Não Funcional
- SIGAA – Sistema Integrado de Gestão de Atividades Acadêmicas
- RU – Restaurante Universitário
- IA – Inteligência Artificial

Descrição geral do sistema

O sistema proposto, denominado "IA do Estudante", é um chatbot desenvolvido para auxiliar alunos universitários no uso da plataforma SIGAA. Seu principal objetivo é oferecer respostas rápidas e úteis às dúvidas mais comuns dos estudantes, como matrícula, trancamento, consulta de histórico escolar, notas, frequência, cadastro no Restaurante Universitário (R.U.) e conteúdos de aulas perdidas.

A aplicação visa facilitar o acesso às informações acadêmicas, reduzindo a sobrecarga de atendimentos aos professores e secretarias, além de tornar a navegação no SIGAA mais intuitiva para os alunos. O chatbot será alimentado com informações previamente inseridas por professores, como materiais didáticos, cronogramas e resumos de aula, permitindo que os estudantes recuperem conteúdos importantes mesmo após faltas.

O sistema será acessível via web, com compatibilidade para computadores e dispositivos móveis, e poderá futuramente ser integrado com plataformas de mensagens. O uso da internet é essencial para seu funcionamento, já que todas as interações ocorrerão em tempo real

Abrangência e sistemas relacionados

O sistema IA do Estudante será capaz de responder a dúvidas frequentes dos alunos sobre processos acadêmicos relacionados ao SIGAA. Entre as funcionalidades principais, destacam-se:

- Consulta de conteúdo de aulas perdidas
- Orientações sobre matrícula, trancamento, histórico e frequência
- Informações sobre o cadastro no Restaurante Universitário (R.U.)
- Direcionamento por meio de links ou explicações passo a passo

O sistema não irá realizar ações dentro do SIGAA, como efetuar matrícula ou editar dados, mas apenas orientar o usuário. Também não substituirá o portal oficial, atuando como um facilitador. O sistema é independente e totalmente auto-contido, mas baseado em informações alimentadas manualmente por professores. Ou seja, os dados exibidos pelo chatbot são inseridos e atualizados pelos docentes em um painel específico, garantindo assim autonomia e flexibilidade.

O chatbot será independente, mas poderá exibir links para o SIGAA. Futuramente, poderá ser integrado a APIs do SIGAA, caso estas estejam disponíveis.

O sistema será acessível via web e mobile, possibilitando consultas rápidas por meio de dispositivos diversos, como celular ou computador.

Descrição dos usuários

1. Aluno

1.1. Os alunos são os principais usuários do chatbot. São eles que interagem diretamente com o sistema, fazendo perguntas sobre procedimentos acadêmicos e buscando conteúdos das aulas perdidas.

1.2. Utiliza o sistema para tirar dúvidas acadêmicas, consultar conteúdos da aula, links úteis e processos como matrícula, histórico, ver notas e faltas, receber instruções sobre trancamento ou cadastro no RU.

2. Professor

2.1 Os professores não utilizam diretamente o chatbot, mas são responsáveis por alimentar o sistema com os conteúdos das aulas ministradas, arquivos, listas de leitura, links e trilhas de estudo. Isso garante que os alunos tenham acesso a informações atualizadas sem precisar contatar os docentes diretamente. Com responsabilidade de inserir informações sobre aulas ministradas, atualizar conteúdos de acordo com o plano de ensino e informar eventuais cancelamentos ou mudanças de cronograma.

Requisitos funcionais

Consultas Acadêmicas

[RF001] Consulta de matrícula

Permite ao aluno verificar o status de sua matrícula em disciplinas do semestre.

Entrada: Nome da disciplina e semestre

Saída: Status da matrícula (deferida ou indeferida ou pendente)

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Aluno

```
# --- CONFIGURAÇÃO DA DISCIPLINA (Área d  Untitled-1 •
1  # --- CONFIGURAÇÃO DA DISCIPLINA (Área do Coordenador/Professor) ---
2
3  # 1. Defina os dados da disciplina
4  disciplina_exemplo = "Inteligência Artificial"
5  semestre_exemplo = "2025.1"
6
7  # 2. Escolha o status da matrícula para esta disciplina.
8  #   As opções são: "Deferida", "Indeferida", "Pendente"
9  #   Basta alterar o texto dentro das aspas abaixo.
10 status_definido_pelo_coordenador = "Deferida"
11
12
13 # --- SISTEMA DE CONSULTA (Área do Aluno) ---
14 # (Nenhuma alteração precisa ser feita daqui para baixo)
15
16 # Recebe os dados do aluno
17 disciplina_digitada = input("Digite o nome da disciplina: ")
18 semestre_digitado = input("Digite o semestre (ex: 2025.1): ")
19
20 # Verifica se a consulta do aluno corresponde à disciplina configurada
21 if disciplina_digitada == disciplina_exemplo and semestre_digitado == semestre_exemplo:
22     # Se a consulta estiver correta, o status final é o que o coordenador definiu
23     status_final = status_definido_pelo_coordenador
24 else:
25     status_final = "Matrícula não encontrada."
26
27 # Exibe o resultado da consulta
28 print(f"\nStatus da Matrícula: {status_final}")
```

• Saída Final:

Status da Matrícula: Deferida

[RF002] Consulta de notas

Permite ao aluno visualizar as suas notas já lançadas no SIGAA, conforme os critérios de avaliação da disciplina.

Entrada: Nome da disciplina

Saída: Lista de avaliações com respectivas notas

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

```
1  # --- Dados do Exemplo (Simulando o que está no SIGAA) ---
2  disciplina_exemplo = "Inteligência Artificial"
3  notas_exemplo = {
4      "Prova 1 (P1)": 8.5,
5      "Projeto Prático": 9.5,
6      "Seminário": 10.0,
7      "Prova 2 (P2)": "Ainda não lançada"
8  }
9
10 # --- Entrada do Aluno ---
11 disciplina_digitada = input("Digite o nome da disciplina para ver as notas: ")
12
13 # --- Verificação e Saída ---
14 # Compara a entrada do usuário com a disciplina do exemplo.
15 if disciplina_digitada == disciplina_exemplo:
16     print(f"\n--- Notas para {disciplina_exemplo} ---")
17     # Itera sobre o dicionário de notas e imprime cada avaliação e sua nota.
18     for avaliacao, nota in notas_exemplo.items():
19         print(f"- {avaliacao}: {nota}")
20 else:
21     # Caso a disciplina não seja encontrada.
22     print(f"\nDisciplina não encontrada ou sem notas lançadas.")
```

• Saída Gerada:

```
--- Notas para Inteligência Artificial ---
- Prova 1 (P1): 8.5
- Projeto Prático: 9.5
- Seminário: 10.0
- Prova 2 (P2): Ainda não lançada
```

[RF003] Emissão de histórico

Permite ao aluno solicitar um histórico simplificado com as disciplinas cursadas e os resultados obtidos.

Entrada: Nome e matrícula

Saída: Histórico acadêmico simplificado

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Aluno

```
# --- Dados do Aluno e seu Histórico ---
1 # --- Dados do Aluno e seu Histórico ---
2 aluno_nome = "Carlos Andrade"
3 aluno_matricula = "202300456"
4
5 disciplinas_cursadas = [
6     {
7         "periodo": "2024.1",
8         "disciplina": "Lógica de Programação",
9         "nota_final": 9.2,
10        "situacao": "Aprovado"
11    },
12    {
13        "periodo": "2024.2",
14        "disciplina": "Estrutura de Dados",
15        "nota_final": 4.0,
16        "situacao": "Reprovado"
17    },
18    {
19        "periodo": "2025.1",
20        "disciplina": "Banco de Dados",
21        "nota_final": 8.8,
22        "situacao": "Aprovado"
23    }
24 ]
25
26 # --- Saída Minimalista ---
27
28 # 1. Informações do aluno em uma única linha
29 print(f"Histórico de {aluno_nome} (Matrícula: {aluno_matricula})\n")
30
31 # 2. Itera sobre a lista e imprime cada disciplina diretamente
32 for disciplina in disciplinas_cursadas:
33     periodo = disciplina['periodo']
34     nome = disciplina['disciplina']
35     nota = disciplina['nota_final']
36     situacao = disciplina['situacao']
37
38     print(f"{periodo:<10} | {nome:<25} | {nota:<6.1f} | {situacao}")
```

Saída Gerada pelo Código:

Histórico de Carlos Andrade (Matrícula: 202300456)

2024.1	Lógica de Programação	9.2	Aprovado
2024.2	Estrutura de Dados	4.0	Reprovado
2025.1	Banco de Dados	8.8	Aprovado

[RF004] Informações sobre RU

Aluno informa se é calouro ou veterano, caso seja veterano irá enviar o link de instruções do funcionamento do RU, caso seja calouro irá enviar um link direcionado para cadastro no sistema, com o passo a passo.

Entrada: Semestre de ingresso, se o Semestre de ingresso for igual ao semestre atual, então é calouro, se for mais antigo do que o semestre atual, então é calouro, se for colocado semestre futuro está inválido

Saída: Link de Instruções ou link para cadastro no RU ou Semestre inválido

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Caso(s) de uso associado(s): Aluno

```
# --- CONFIGURAÇÃO DO SISTEMA ---
1 # --- CONFIGURAÇÃO DO SISTEMA ---
2 # Com base na data atual (29 de Julho de 2025), o semestre vigente é o primeiro.
3 semestre_atual = "2025.1"
4
5 # Links do Restaurante Universitário (RU)
6 link_instrucoes_veteranos = "http://unifap.br/ru/instrucoes-2025"
7 link_cadastro_calouros = "http://unifap.br/ru/cadastro-calouros-2025.1"
8
9
10 # --- INTERAÇÃO COM O ALUNO ---
11 print("--- Sistema de Acesso ao Restaurante Universitário (RU) ---")
12 semestre_ingresso = input("Digite seu semestre de ingresso (formato AAAA.S, ex: 2024.2): ")
13
14
15 # --- PROCESSAMENTO E SAÍDA ---
16
17 # Primeiro, uma validação simples para o formato da entrada
18 if "." not in semestre_ingresso or len(semestre_ingresso) != 6:
19     print("\n[ERRO] Formato inválido. Por favor, use o formato AAAA.S, como em '2024.1'.")
20
21 # Compara o semestre de ingresso com o semestre atual
22 elif semestre_ingresso > semestre_atual:
23     print("\n[ERRO] Semestre inválido.")
24     print("Você digitou um semestre futuro. Por favor, verifique o valor inserido.")
25
26 elif semestre_ingresso == semestre_atual:
27     print("\nBem-vindo(a), Calouro(a)!")
28     print("Para usar o RU, você precisa primeiro realizar seu cadastro no sistema.")
29     print(f"Acesse o link com o passo a passo aqui: {link_cadastro_calouros}")
30
31 else: # Se não é futuro nem igual, só pode ser mais antigo (veterano)
32     print("\nOlá, Veterano(a)!")
33     print("Para informações sobre horários, cardápio e regras, acesse nosso guia.")
34     print(f"Link de instruções atualizadas: {link_instrucoes_veteranos}")
```

Cenário 1: Aluno Calouro

- Entrada do usuário: 2025.1
- Saída Gerada:

```
Bem-vindo(a), Calouro(a)!
Para usar o RU, você precisa primeiro realizar seu cadastro no sistema.
Acesse o link com o passo a passo aqui: http://unifap.br/ru/cadastro-calouros-2025.1
```

Cenário 2: Aluno Veterano

- Entrada do usuário: 2023.2
- Saída Gerada:

```
Olá, Veterano(a)!
Para informações sobre horários, cardápio e regras, acesse nosso guia.
Link de instruções atualizadas: http://unifap.br/ru/instrucoes-2025
```

Cenário 3: Semestre Futuro (Inválido)

- Entrada do usuário: 2026.1
- Saída Gerada:

```
[ERRO] Semestre inválido.
Você digitou um semestre futuro. Por favor, verifique o valor inserido.
```

[RF005] Acesso ao conteúdo de aula perdida

Permite ao aluno perguntar o que foi dado em uma aula que perdeu.

Entrada: Nome da disciplina e data da aula, se o professor disponibilizou o que foi dado na aula, então a informação será válida, caso não esteja disponível, a informação será inválida

Saída: Conteúdo (resumo, link de slides, referências) da aula específica ou conteúdo da aula indisponível

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Aluno

```
# --- Exemplo de um único conteúdo de aula ---
1 # --- Exemplo de um único conteúdo de aula disponibilizado ---
2 disciplina_exemplo = "Inteligência Artificial"
3 data_exemplo = "2025-07-22"
4 conteudo_exemplo = """- Resumo: Introdução aos Algoritmos de Busca (BFS e DFS).
5 - Slides: http://unifap.br/ia/aula5-slides.pdf"""
6
7
8 # --- Entrada do Aluno ---
9 disciplina_digitada = input("Disciplina: ")
10 data_digitada = input("Data da aula (AAAA-MM-DD): ")
11
12
13 # --- Verificação e Saída Direta ---
14 # Compara se a entrada do aluno corresponde exatamente ao exemplo disponível.
15 if disciplina_digitada == disciplina_exemplo and data_digitada == data_exemplo:
16     # Se sim, exibe o conteúdo da aula
17     print("\n-- Conteúdo da Aula --")
18     print(conteudo_exemplo)
19 else:
20     # Se não, informa que o conteúdo não está disponível
21     print("\nConteúdo da aula indisponível.")
```

• Interação com o usuário:

```
Disciplina: Inteligência Artificial
Data da aula (AAAA-MM-DD): 2025-07-22
```

• Saída Gerada:

```
-- Conteúdo da Aula --
- Resumo: Introdução aos Algoritmos de Busca (BFS e DFS).
- Slides: http://unifap.br/ia/aula5-slides.pdf
```

ou

• Interação com o usuário:

```
Disciplina: Banco de Dados
Data da aula (AAAA-MM-DD): 2025-07-22
```

• Saída Gerada:

```
Conteúdo da aula indisponível.
```

[RF006] Trilha de estudo para prova

Permite ao aluno solicitar uma trilha de estudo para se preparar para uma avaliação.

Entrada: Nome da disciplina e data da prova

Saída: Lista de conteúdos cobrados e materiais indicados, caso não tenha prova para a data digitada ou esteja errada, coloque que não foi encontrada

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Caso(s) de uso associado(s): Aluno

```
# --- Exemplo Base --- Untitled-1
1 # --- Exemplo Base ---
2 disciplina_exemplo = "Estrutura de Dados"
3 data_prova_exemplo = "2025-08-15"
4 trilha_exemplo = "Conteúdos: Pilhas, Filas, Listas Encadeadas.\nMateriais: Slides 4-7 e Lista de Exercícios."
5
6
7 # --- Entrada do Aluno ---
8 disciplina_digitada = input("Disciplina: ")
9 data_digitada = input("Data da prova (AAAA-MM-DD): ")
10
11
12 # --- Lógica e Saída ---
13 if disciplina_digitada == disciplina_exemplo and data_digitada == data_prova_exemplo:
14     print("\n--- Trilha de Estudo ---\n" + trilha_exemplo)
15 else:
16     print("\nTrilha de estudo não encontrada.")
```

Cenário 1: Entrada Correta

Se o usuário digitar `Estrutura de Dados` e `2025-08-15`:

- Interação:**

```
Disciplina: Estrutura de Dados
Data da prova (AAAA-MM-DD): 2025-08-15
```

- Saída Gerada:**

```
--- Trilha de Estudo ---
Conteúdos: Pilhas, Filas, Listas Encadeadas.
Materiais: Slides 4-7 e Lista de Exercícios.
```

Cenário 2: Entrada Incorreta

Se o usuário digitar qualquer outra informação:

 - Interação:**

```
Disciplina: Cálculo I
Data da prova (AAAA-MM-DD): 2025-08-15
```

 - Saída Gerada:**

```
Trilha de estudo não encontrada.
```

Requisitos não funcionais

Usabilidade

[NF001] Linguagem diferente para com Docente e Discente

O chatbot deve utilizar linguagem simples, direta e menos formal com o aluno. Com o professor a linguagem deverá ser totalmente formal, utilizando-se do jargão.

Métrica: Chatbot terá um limite de 80 caracteres para responder ao aluno, para responder o professor terá uma resposta limite de 180 caracteres

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

Confiabilidade

[NF002] Coerência da Informação Fornecida

As informações devem ser sempre consistentes com os dados inseridos pelos professores.

Métrica: Relatório semanal em pdf com divergências detectadas por comparação de informações, as enviadas pelo professor e as encontradas no banco de dados.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

[NF003] Disponibilidade do Serviço

O chatbot deve estar disponível 24 horas por dia, 6 dias por semana, exceto durante o domingo, dia de manutenção que deverá ser sempre previamente comunicado, com duração de no máximo 12 horas.

Métrica: Tempo de inatividade (downtime) máximo de 48 horas mensais. Contendo uma planilha em excel realizando o controle do tempo.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

[NF004] Recuperação de Falhas

Em caso de falha do sistema, o chatbot deve ser capaz de restaurar a operação e os dados para o último ponto de salvamento consistente em até 30 minutos.

Métrica: Tempo de recuperação (RTO) ≤ 30 minutos após a detecção da falha. Com uma mensagem pedindo desculpa pela falha e solicitando que o usuário realize novamente seu pedido.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

Desempenho

[NF005] Tempo de Resposta Rápido

As respostas do chatbot devem ser geradas em até 1 minuto após a entrada do usuário, sendo isso para pedidos mais complexos que envolvam cálculo matemático de segundo grau, para perguntas mais simples a resposta deverá ser gerada em até 20 segundos.

Métrica: Interações com mais até 1 pedido ou 3 pedidos simples deverá ter 90% de solidez com o tempo determinado de 20 segundos. Interações 1 ou mais pedidos complexos deverão ter 75% de solidez com o tempo determinado de 1 minuto.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

[NF006] Suporte a Usuários Concomitantes

O sistema deve ser capaz de suportar pelo menos 150 usuários simultâneos sem degradação perceptível no tempo de resposta e queda do sistema

Métrica: Testes de carga com 100 usuários simultâneos semanalmente, mantendo o NF003.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

[NF007] Otimização do Consumo de Recursos

O consumo de CPU e memória do servidor do chatbot não deve exceder 90% da capacidade total durante picos de uso.

Métrica: Monitoramento de recursos do servidor, com alertas no terminal para uso acima de 80% e alertando para um possível pico acima de 90%.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

Segurança

[NF008] Gerenciamento de Sessão:

As sessões de usuário devem expirar automaticamente após 30 minutos de inatividade.

Métrica: Após o login haverá um cronômetro que irá fazer contagem regressiva dos 30 minutos após inatividade da tela.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

Distribuição

[NF009] Multiplataforma

O sistema deve estar acessível tanto por navegador (web) quanto por dispositivo móvel (mobile).

Métrica: Compatibilidade comprovada por testes em navegadores (Chrome, Firefox, Edge) e sistemas operacionais móveis (Android/iOS).

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

Padrões

[NF010] Aderência a Padrões de Acessibilidade

A interface deve seguir boas práticas de acessibilidade, como uso de contraste, botões visíveis e suporte a leitores de tela.

Métrica: Avaliação baseada nas diretrizes WCAG 2.1 (Web Content Accessibility Guidelines).

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos

Hardware e software

[NF011] Ambiente de Produção Contratado

O ambiente de hospedagem e execução do chatbot em produção deve ser um serviço de nuvem contratado e gerenciado, como AWS, Google Cloud ou Azure, garantindo escalabilidade, segurança e alta disponibilidade.

Métrica: Contrato de nível de serviço (SLA) do provedor de nuvem que garanta 99,9% de uptime e relatório mensal de utilização dos recursos.

Prioridade: ☐ Essencial ☐ Importante ☐ Desejável

Caso(s) de uso associado(s): Todos