



Product Backlog – AUXIA Project

API 6º ADS

Histórico das Revisões

Data	Versão	Descrição	Autores
25/02/2025	V-001	Criação do Documento	Ivan G. R. O. Duarte
01/04/2025	V-002	Adição do DoD e DoR	Ivan G. R. O. Duarte
02/04/2025	V-003	Revisão	Ivan G. R. O. Duarte
05/04/2025	V-004	Adição do Planning da Sprint 2	Ivan G. R. O. Duarte
07/04/2025	V-005	Ajustes do Backlog – Sprint 2	Ivan G. R. O. Duarte
30/04/2025	V-006.0	Adição do Planning da Sprint 3 (INCOMPLETO)	Ivan G. R. O. Duarte
12/05/2025	V-006.1	Atualização da Sprint 3	Ivan G. R. O. Duarte
22/05/2025	V-007	Remoção de User Stories descontinuadas da tabela do Backlog total do produto.	Ivan G. R. O. Duarte

SUMÁRIO

1	INTRODUÇÃO.....	5
2	OBJETIVOS DO DOCUMENTO.....	6
3	CONVENÇÕES VISUAIS UTILIZADAS NO BACKLOG.....	7
4	REQUISITOS DO CLIENTE – API 6º SEMESTRE ADS.....	8
5	PRODUCT BACKLOG – AUXIA.....	9
6	SPRINT BACKLOG.....	12
6.1	SPRINT 01.....	12
6.1.1	DoD & DoR – Sprint 1.....	13
6.1.1.1	Definition Of Ready (DOR) – Sprint 1.....	13
6.1.1.2	Definition Of Done (DOD) – Sprint 1.....	13
6.1.2	Cenários E Critérios De Aceite – User Stories – Sprint 1.....	14
6.1.2.1	US #01: <i>Interface para Inserção de Prompt</i>	14
6.1.2.2	US #02: <i>Envio Simultâneo de Prompt para Múltiplas LLMs</i>	15
6.1.2.3	US #04: <i>Avaliação Individual das Respostas com Critérios</i>	18
6.1.2.4	US #05: <i>Escolha da Melhor Resposta e Justificativa</i>	21
6.1.2.5	US #06: <i>Armazenamento das Avaliações para Fine-Tuning</i>	23
6.1.2.6	US #10: <i>Administrador Pré-Cadastrado no Sistema</i>	25
6.2	SPRINT 02.....	26
6.2.1	DoD & DoR – Sprint 2.....	27
6.2.1.1	Definition Of Ready (DOR) – Sprint 2.....	27
6.2.1.2	Definition Of Done (DOD) – Sprint 2.....	28
6.2.2	Cenários E Critérios De Aceite – User Stories – Sprint 2.....	29

6.2.2.1	US #07: <i>Alerta de Inconsistência na Escolha da Resposta</i>	29
6.2.2.2	US #08: <i>Enriquecimento das Respostas com Base Vetorizada - RAG Parte 1</i>	30
6.2.2.3	US #15: <i>Revisão da Escolha Antes da Submissão</i>	31
6.2.2.4	US #19: <i>Alerta de Timeout no Envio ou Resposta das LLMS</i>	33
6.2.2.5	US #21: <i>Confirmação que a Avaliação foi Registrada - Status</i>	34
6.2.2.6	US #22: <i>Navegação para Telas Anteriores Durante a Avaliação</i>	36
6.3	SPRINT 03.....	38
6.3.1	DoD & DoR – Sprint 3.....	38
6.3.1.1	Definition Of Ready (DOR) – Sprint 3.....	39
6.3.1.2	Definition Of Done (DOD) – Sprint 3.....	39
6.3.2	Cenários E Critérios De Aceite – User Stories – Sprint 2.....	40
6.3.2.1	US #11: <i>Cadastro de Novos Usuários na Plataforma</i>	40
6.3.2.2	US #12: <i>Interface de Login para Acesso ao Sistema</i>	42
6.3.2.3	US #13: <i>Autenticação de Usuários Autorizados</i>	43
6.3.2.4	US #14: <i>Logout Seguro da Aplicação</i>	44
6.3.2.5	US #16: <i>Visualização e Gestão da Lista de Usuários</i>	45
6.3.2.6	US #17: <i>Redefinição de Senha para Usuários como Administrador</i>	47
6.3.2.7	US #18: <i>Exclusão de Usuários pelo Administrador</i>	48
6.3.2.8	US #24: <i>Edição de Dados Pessoais</i>	49
7	USER STORIES DESCONTINUADAS OU INCORPORADAS.....	51
7.1	US #00: <i>OBTENÇÃO DE RESPOSTAS DAS LLMS</i>	51
7.2	US #23: <i>PRESERVAÇÃO DO PROGRESSO AO RETORNAR PARA TELAS ANTERIORES</i>	51
7.3	US #09: <i>GERAÇÃO DE RESPOSTAS BASEADAS NO CONTEXTO RECUPERADO</i>	52
7.4	US #20: <i>ALERTA DE ERRO NA OBTENÇÃO DAS RESPOSTAS</i>	53
8	CONSIDERAÇÕES FINAIS.....	54

1 INTRODUÇÃO

Este documento tem como objetivo apresentar o Backlog do Produto da solução proposta para avaliação de modelos de linguagem (LLMs) com base em feedback humano estruturado. O backlog foi construído considerando as necessidades do cliente (Dom Rock), os requisitos funcionais e não funcionais levantados, e os objetivos do projeto, com base nas práticas recomendadas de desenvolvimento ágil.

Através do processo de levantamento e refinamento de requisitos, foram definidas User Stories organizadas em Épicas, representando as funcionalidades essenciais para garantir a usabilidade, escalabilidade e evolução do sistema. Cada User Story está descrita em um formato claro e objetivo, utilizando o padrão:

“Como [persona], quero [ação] para que [objetivo]”

Além disso, este backlog contempla a divisão de responsabilidades entre frontend e backend, a validação lógica de interações críticas, o armazenamento estratégico de dados para uso em fine-tuning, e a preparação da arquitetura para futuras integrações com RAG (Retrieval-Augmented Generation).

Este documento servirá como guia para o planejamento das Sprints, priorização de entregas com técnicas como MoSCoW e Matriz de Eisenhower, além de apoiar a comunicação entre as partes envolvidas durante o desenvolvimento do projeto.

2 OBJETIVOS DO DOCUMENTO

Este documento tem como objetivo registrar e estruturar todas as User Stories que compõem a Sprint 1 do projeto, de forma clara, objetiva e alinhada às necessidades do cliente. Ele serve como base para o desenvolvimento das funcionalidades priorizadas, facilitando o entendimento compartilhado entre o Product Owner, equipe técnica e stakeholders. Além disso, o documento contribui para o planejamento, rastreamento e validação das entregas, apoiando o time na construção incremental e iterativa da solução.

- Organizar e detalhar as funcionalidades que compõem o produto;
- Garantir clareza e alinhamento entre stakeholders e equipe de desenvolvimento;
- Servir como base para refinamento, priorização e planejamento de iterações ágeis;
- Registrar as decisões iniciais e evoluções do escopo do sistema.

3 CONVENÇÕES VISUAIS UTILIZADAS NO BACKLOG

Para facilitar a leitura e a manutenção do histórico de decisões durante o desenvolvimento do projeto, este documento adota a seguinte *convenção visual* para identificar o status de itens que foram descontinuados.

Estilo Visual	Cor	Significado
Fundo Branco (padrão)		Requisitos levantados e acordados entre o PO e Cliente User Stories ativas — estão planejadas, em desenvolvimento ou já entregues.
Fundo Laranja Pastel		Itens que foram removidos do escopo atual por decisão da equipe durante o planejamento das Sprints. Permanecem no documento para manter o histórico de evolução do produto.

Nota: Cada US descontinuada possui uma justificativa na seção "User Stories Descontinuadas ou Incorporadas", indicando o motivo da decisão e, quando aplicável, em qual US ela foi incorporada.

4 REQUISITOS DO CLIENTE – API 6º SEMESTRE ADS

ID	TIPO	Título	Descrição
R01	Funcional	Integração com 2 APIs públicas de LLMs.	O sistema deve permitir conexão simultânea com dois modelos de IA via API.
R02	Funcional	Tela de interface para obtenção dos prompts e submissão às LLMs	O usuário deve poder inserir um prompt e enviá-lo para os modelos de IA.
R03	Funcional	Tela de interface para obtenção de feedback humano com avaliação de critérios das respostas da LLM1 e LLM2	O usuário deve avaliar as respostas com critérios e escrever a justificativa da nota do critério através de um texto.
R04	Funcional	Tela de interface para escolha final entre LLM1 e LLM2	O usuário deve poder escolher entre a resposta da LLM1 e LLM2, porém é necessário que essa escolha seja coerente com os critérios escolhidos na tela anterior.
R05	Funcional	Persistência das avaliações do RLHF	O sistema deve armazenar as avaliações dos usuários em um banco de dados estruturado.
R06	Funcional	Autenticação e gestão de Usuários	O sistema deve contar com um painel para o administrador cadastrar novos usuários e contar também com usuários comuns para utilização da aplicação.
R07	Funcional	Implementação do RAG para melhoria do RLHF	O sistema deverá ser capaz de utilizar o RAG para buscar informações em uma base de dados externa, melhorando a qualidade das respostas dos LLMs.
R08	Não Funcional	Vídeo tutorial	O cliente gostaria de um vídeo demonstrativo para que, usuários sem conhecimento técnico da plataforma, consigam fazer uma avaliação seguindo os passos do vídeo.
R09	Não Funcional	Uso de modelos LLM de acesso público	O sistema deve utilizar APIs de LLMs de domínio público, tais como: Hugging Face, Gemini, Grok, Llama, OpenAI.
R10	Funcional	Gráficos de desempenho das LLMs	OBS: O cliente descartou a necessidade desse requisito em conversa com o P.O formalizada via aplicação de mensagens "Slack".

5 PRODUCT BACKLOG – AUXIA

Rank	Prioridade	User Story	Story Points	Sprint	Requisito do Cliente	Status
1	Alta	Como usuário, quero uma interface para inserir um prompt, para que eu possa enviá-lo às LLMs e obter suas respostas	14	1	R02	✓
2	Alta	Como usuário, eu quero enviar um prompt para dois modelos de IA simultaneamente, para que eu possa avaliar suas respostas posteriormente.	47	1	R01/R09	✓
3	Alta	Como usuário, eu quero visualizar as respostas das LLMs de forma clara e acessível, para que eu possa analisá-las em critérios.	28	1	R03	✓
4	Alta	Como usuário, eu quero uma interface para poder avaliar cada resposta individualmente através de critérios definidos, para que eu possa analisar a qualidade das respostas geradas.	22	1	R03	✓
5	Alta	Como usuário, eu quero uma interface para poder escolher a melhor resposta entre as duas geradas pelas LLMs, para que o sistema registre minha decisão e justificativa.	29	1	R04	✓
6	Alta	Como Administrador, eu quero que as avaliações dos usuários sobre as LLMs sejam armazenadas em um BD, para que possam ser utilizadas em processos de fine-tuning futuramente.	41	1	R05	✓
7	Alta	Como usuário, eu quero ser alertado caso minha escolha entre LLM1 e LLM2 não esteja coerente com minha avaliação, para que eu possa revisar minha decisão antes de finalizar.	3	2	R04	✓
8	Alta	Como usuário, eu quero que as respostas das LLMs sejam enriquecidas com informações relevantes da base de dados (vetorizada), para que sejam mais precisas.	5	2	R07	✓
10	Alta	Como Administrador, eu gostaria de ser o primeiro usuário do sistema, já devidamente pré inserido no banco de dados, para que possa acessar a aplicação.	44	1	R06	✓
11	Média	Como Administrador, eu quero cadastrar novos usuários na plataforma, para que somente pessoas autorizadas possam acessá-la.	N/D	3	R06	✓

12	Média	Como usuário, eu quero acessar a aplicação através de uma interface de login, para que somente usuários autorizados possam utilizar o sistema.	N/D	3	R06	✓
13	Média	Como usuário autorizado, eu quero acessar o sistema através de um login, para utilizar a aplicação.	N/D	3	R06	✓
14	Média	Como usuário autenticado, eu quero poder fazer o logout da aplicação de forma segura, para que meus dados não fiquem acessíveis a terceiros.	N/D	3	R06	✓
15	Média	Como usuário, eu quero poder revisar minha escolha antes de submetê-la, para que eu tenha certeza de que minha decisão está correta.	2	2	R04	✓
16	Média	Como Administrador, eu quero visualizar a lista de usuários cadastrados, para que eu possa gerenciar quem tem acesso ao sistema.	N/D	3	R06	✓
17	Média	Como Administrador, eu quero redefinir a senha de um usuário, para que eu possa ajudá-lo caso ele não consiga acessar a conta.	N/D	3	R06	✓
18	Média	Como Administrador, eu quero excluir usuários do sistema, para que possa revogar o acesso de usuários a aplicação.	N/D	3	R06	✓
19	Baixa	Como usuário, eu quero ser informado com mensagens de erro caso ocorra demora excessiva no envio do prompt ou na resposta das LLMs, ou outros erros, para que eu possa entender o problema e tentar novamente.	2	2	R03	✓
21	Baixa	Como usuário, eu quero poder receber mensagens claras sobre o status das avaliações, para ter certeza de que minha avaliação foi registrada corretamente	2	2	R03	✓
22	Baixa	Como usuário, eu quero poder voltar para telas anteriores durante o processo de avaliação, para que eu possa corrigir informações antes de enviar a decisão final.	1	2	R03/R04	✓
24	Baixa	Como usuário, eu quero editar meus dados pessoais, para que eu possa manter minhas informações atualizadas.	N/D	3	R06	✓

6 SPRINT BACKLOG

6.1 SPRINT 01

Rank	Prioridade	User Story	Story Points	Sprint	Requisito do Cliente	Status
1	Alta	Como usuário, quero uma interface para inserir um prompt, para que eu possa enviá-lo às LLMs e obter suas respostas	14	1	R02	✓
2	Alta	Como usuário, eu quero enviar um prompt para dois modelos de IA simultaneamente, para que eu possa avaliar suas respostas posteriormente.	47	1	R01/R09	✓
3	Alta	Como usuário, eu quero visualizar as respostas das LLMs de forma clara e acessível, para que eu possa analisá-las em critérios.	28	1	R03	✓
4	Alta	Como usuário, eu quero uma interface para poder avaliar cada resposta individualmente através de critérios definidos, para que eu possa analisar a qualidade das respostas geradas.	22	1	R03	✓
5	Alta	Como usuário, eu quero uma interface para poder escolher a melhor resposta entre as duas geradas pelas LLMs, para que o sistema registre minha decisão e justificativa.	29	1	R04	✓
6	Alta	Como Administrador, eu quero que as avaliações dos usuários sobre as LLMs sejam armazenadas em um BD, para que possam ser utilizadas em processos de fine-tuning futuramente.	41	1	R05	✓
10	Alta	Como Administrador, eu gostaria de ser o primeiro usuário do sistema, já devidamente pré inserido no banco de dados, para que possa acessar a aplicação.	44	1	R06	✓

6.1.1 DoD & DoR – Sprint 1

6.1.1.1 Definition Of Ready (DOR) – Sprint 1

Critério	Descrição
Clareza na Descrição	A User Story está escrita no formato “Como [persona], quero [ação] para que [objetivo]”.
Critérios de Aceitação Definidos	A história possui critérios objetivos que indicam o que é necessário para considerá-la concluída.
Cenários de Teste Especificados	A história tem pelo menos 1 cenário de teste estruturado (Dado, Quando, Então).
Independente	A história pode ser implementada sem depender de outra tarefa da mesma Sprint.
Compreensão Compartilhada	Toda a equipe (incluindo PO e devs) compreende o propósito da história.
Estimável	A história foi pontuada no Planning Poker ou tem uma estimativa clara.
Documentação de Apoio	Se necessário, mockups, fluxos ou modelos de dados estão anexados ou referenciados.
Critérios técnicos acordados	As necessidades de Frontend e Backend foram claramente separadas (quando aplicável).

6.1.1.2 Definition Of Done (DOD) – Sprint 1

Critério	Descrição
Critérios de Aceitação atendidos	Todos os critérios da US foram validados pelo time e passaram nos testes.

Testes manuais realizados	Todos os cenários de teste da história foram executados e aprovados.
Persistência dos dados garantida	Onde aplicável (ex: US07), os dados são corretamente armazenados e recuperáveis.
Código revisado (Code Review)	O código foi revisado por pelo menos um colega de equipe.
Documentação interna atualizada	Foi atualizado o que for necessário: API, estrutura de dados, endpoints, etc.
Integração com outras partes testada	As interfaces entre Frontend e Backend foram validadas.
Build/testes automatizados (se aplicável)	A funcionalidade não quebra a aplicação e passa nos testes automatizados existentes.
Validação do PO	O Product Owner validou a entrega com base nos critérios definidos.
Pronto para deploy	O item está testado, validado e pode ser integrado ao produto final.


6.1.2 Cenários E Critérios De Aceite – User Stories – Sprint 1

6.1.2.1 US #01: *Interface para Inserção de Prompt*

Descrição:

Comentário PO:

– “Esta User Story é relacionada ao FrontEnd, diz respeito a interface que o usuário irá encontrar no primeiro contato com o RLHF de nossa aplicação.”

– “É da responsabilidade do  Dev Team (@Pedro Henrique Davi Gonçalves @Jonas Alves @Isaque Elis da Silva @Vitor Lima @Rafael Motta @Joice Aparecida de Araújo), de combinar quem será o responsável pela integração das funcionalidades, neste caso, o “Botão de Enviar o ‘Prompt’.”

User Story:

“Como usuário, quero uma interface para inserir um prompt, para que eu possa enviá-lo às LLMs e obter suas respostas.”

Critério de Aceitação:

- A aplicação deve exibir um campo de entrada onde o usuário pode digitar um prompt.
- O botão "Enviar" deve estar disponível e ativo após a digitação do prompt e inativo quando não for inserido nenhum texto.
- Após o clique no botão de “Enviar”, a aplicação deve direcionar o conteúdo do prompt para a função/rota responsável por direcioná-lo às LLMs
- Após o envio ser bem sucedido, a aplicação deverá direcionar o usuário a tela de avaliação das respostas.
- A interface deve seguir os padrões estabelecidos no Wireframe disponível no Figma: [LINK](#)

Cenários:

Cenário 1 – Envio de Prompt com Sucesso

- Dado que o usuário acessa a tela de inserção de prompt,
- Quando ele digita um prompt válido e clica no botão "Enviar",
- Então o sistema deve encaminhar o prompt para as LLMs.

Cenário 2 – Tentativa de Envio com Campo Vazio

- Dado que o usuário acessa a tela de inserção de prompt,
- Quando ele tenta enviar um prompt sem digitar nada,
- Então o sistema deve desabilitar o botão "Enviar".

Cenário 3 – Erro na Comunicação com as LLMs

- Dado que o usuário insere um prompt e clica em "Enviar",
- Quando houver uma falha na comunicação com as APIs das LLMs,
- Então o sistema deve exibir uma mensagem de erro e permitir uma nova tentativa.

6.1.2.2 US #02: Envio Simultâneo de Prompt para Múltiplas LLMs

Descrição:

Comentário PO:

– “Esta User Story está relacionada ao encapsulamento da lógica do BackEnd, considerando a requisição que a aplicação fará as duas LLMs para enviar o Prompt do Usuário. Também é responsável por encapsular a lógica do BackEnd referente ao momento da obtenção de respostas por parte dos LLMs, e o envio das mesmas ao FrontEnd.”

– “Esta User Story foi unida com a US “Obtenção de Respostas das LLMs” após o Dev Team formar maioria favorável a junção.”

User Story:

“Como usuário, eu quero enviar um prompt para dois modelos de IA simultaneamente, para que eu possa avaliar suas respostas posteriormente.”

Critério de Aceitação:

- O sistema deve estabelecer uma rota de conexão com duas APIs de LLMs de uso público
- O sistema deve permitir o envio de um único prompt para duas LLMs ao mesmo tempo.
- As respostas das LLMs devem ser obtidas e retornadas ao serviço que fez a requisição.
- O sistema deve receber a resposta do prompt enviado as LLMs.
- As respostas das LLMs devem ser processadas e preparadas para serem retornadas ao FrontEnd.
- Se o serviço que espera as respostas, não obtiver retorno das LLMs, o sistema deve registrar um erro e cancelar a operação.

Cenários:

Cenário 1 – Envio bem-sucedido para ambas as LLMs

- Dado que o sistema recebe um prompt válido,
- Quando ele encaminha o prompt para as duas LLMs,
- Então as LLMs recebem o Prompt e iniciam o processo para gerar as respostas.

Cenário 2 – Falha na comunicação com ambas as LLMs

- Dado que o sistema recebe um prompt válido,
- Quando nenhuma das LLMs retorna uma resposta dentro do tempo limite,
- Então o sistema deve retornar um erro informando a falha total na comunicação.

Cenário 3 – Obtenção bem-sucedida de ambas as LLMs

- Dado que o sistema recebe a resposta das duas LLMs,

- Quando ele encaminha a resposta para o serviço responsável por prepará-las para o frontend.
- Então o sistema envia as duas respostas para o FrontEnd.

US #03: Interface para Visualização Clara e Acessível das Respostas

Descrição:

Comentário PO:

– “Esta User Story se refere as necessidades do FrontEnd ao receber as duas respostas das LLMs.”

User Story:

“Como usuário, eu quero visualizar as respostas das LLMs de forma clara e acessível, para que eu possa analisá-las em critérios.”

Critério de Aceitação:

- O sistema deve exibir as respostas das LLMs assim que forem recebidas do backend.
- As respostas das duas LLMs devem ser exibidas lado a lado para facilitar a comparação.
- O sistema deve exibir um indicador de carregamento enquanto aguarda as respostas das LLMs.
- O sistema deve armazenar as respostas recebidas localmente (LocalStorage, Cache/useState), para serem registradas posteriormente em um BD.
- Se uma ou ambas as LLMs falharem, o sistema deve exibir uma mensagem informando que não foi possível obter as respostas.
- Deve haver um botão de reenvio caso a obtenção das respostas falhe.
- A interface deve seguir os padrões estabelecidos no Wireframe disponível no Figma: [LINK](#)

Cenários:

Cenário 1 – Exibição bem-sucedida das respostas das LLMs

- Dado que o usuário acessa a tela de exibição de respostas,
- Quando as respostas das duas LLMs são recebidas corretamente,
- Então o sistema deve exibi-las lado a lado de forma clara e organizada.

Cenário 2 – Exibição de carregamento enquanto aguarda respostas

- Dado que o usuário acessa a tela de exibição de respostas,
- Quando o sistema ainda está aguardando as respostas das LLMs,

- Então um indicador de carregamento deve ser exibido.

Cenário 3 – Falha na obtenção das respostas de ambas as LLMs

- Dado que o usuário acessa a tela de exibição de respostas,
- Quando nenhuma das LLMs retorna uma resposta ou somente uma retorna,
- Então o sistema deve exibir uma mensagem de erro informando que não foi possível obter respostas.

Cenário 4 – Reenvio de prompt após falha

- Dado que o usuário acessa a tela de exibição de respostas,
- Quando a obtenção das respostas falha,
- Então o sistema deve exibir um botão de "Tentar Novamente" para reenviar o prompt.

Cenário 5 – Armazenamento das Respostas no Banco de Dados

- Dado que o sistema recebe um prompt e obtém respostas de ambas as LLMs,
- Quando os dados são processados,
- Então o sistema deve armazenar essas duas respostas localmente no navegador, para posteriormente serem persistidos em um banco de dados.

6.1.2.3 US #04: Avaliação Individual das Respostas com Critérios

Descrição:

Comentário PO:

– “Esta User Story é referente ao FrontEnd e BackEnd desta funcionalidade.

Frontend: O usuário deve poder clicar em uma das respostas, e um modal deve ser exibido centralizando a resposta escolhida e mostrando os critérios pré-definidos com uma escala de 1 a 5. O usuário também deve poder justificar cada critério em um campo de texto limitado a 500 caracteres.

Backend: O sistema deve armazenar a pergunta, a resposta selecionada e as notas de cada critério, garantindo a persistência dos dados para futuras análises.”

– “Conforme apuração com o cliente, temos os seguintes critérios para nos guiar:

Critérios Pré-Definidos:

- **Aderência ao Prompt** – A resposta está diretamente relacionada à intenção do usuário? O modelo compreendeu corretamente as instruções do prompt e respondeu de forma pertinente, sem fugir do tema ou omitir comandos importantes?
- **Coerência e Clareza** – A resposta é bem estruturada, gramaticalmente correta e de fácil compreensão?
- **Exatidão e Confiabilidade (Veracidade da Resposta)** – As informações apresentadas são corretas e baseadas em fontes confiáveis?
- **Exposição e Justificativa** – A resposta fornece explicações ou justificativas adequadas para embasar seu conteúdo?
- **Idioma da Resposta é o Mesmo da Pergunta?** – A resposta foi gerada no idioma correto, conforme solicitado no prompt?
- **Resposta Agressiva ou Ofensiva?** – A resposta contém linguagem inadequada, ofensiva, preconceituosa ou potencialmente prejudicial?

User Story:

“Como usuário, eu quero uma interface para poder avaliar cada resposta individualmente através de critérios definidos, para que eu possa analisar a qualidade das respostas geradas.”

Critério de Aceitação:

FrontEnd:

- O usuário deve poder clicar em uma resposta para abrir um modal de avaliação.
- O modal deve exibir a resposta selecionada centralizada na tela.
- O usuário deve visualizar os critérios pré-definidos para avaliação.
- Cada critério deve permitir a seleção de uma nota em uma escala de 1 a 5.
- Deve haver um campo de justificativa textual para cada critério, limitado a 500 caracteres.
- O modal deve ter um botão "Confirmar Avaliação", que só será habilitado após todos os critérios serem preenchidos.
- O sistema só deverá prosseguir para a tela de “Escolha Final” após as duas respostas serem avaliadas em seus critérios.
- A interface deve seguir os padrões estabelecidos no Wireframe disponível no Figma: [LINK](#)

BackEnd:

- O sistema deve armazenar a pergunta e a resposta selecionada.
- O sistema deve armazenar as notas atribuídas a cada critério.
- O sistema deve armazenar as justificativas inseridas pelo usuário.

- O sistema deve garantir a integridade dos dados e validar que todas as notas e justificativas foram preenchidas antes de salvar.

Cenários:

Cenário 1 – Abrir modal de avaliação ao selecionar uma resposta

- Dado que o usuário está visualizando as respostas das LLMs,
- Quando ele clica em uma das respostas,
- Então um modal deve ser exibido, centralizando a resposta e apresentando os critérios de avaliação.

Cenário 2 – Avaliação bem-sucedida

- Dado que o usuário abriu o modal de avaliação,
- Quando ele preenche todas as notas e justificativas,
- Então o botão "*Confirmar Avaliação*" deve ser habilitado.
- E quando o usuário clicar em "*Confirmar Avaliação*",
- Então o sistema deve atualizar o estado do componente daquela resposta.

Cenário 3 – Tentativa de envio sem preencher todos os critérios

- Dado que o usuário abriu o modal de avaliação,
- Quando ele tenta confirmar a avaliação sem preencher todas as notas ou justificativas,
- Então o sistema deve exibir um aviso informando que todos os critérios devem ser preenchidos.

Cenário 4 – Validação do limite de caracteres na justificativa

- Dado que o usuário está preenchendo a justificativa de um critério,
- Quando ele tenta digitar mais de 500 caracteres,
- Então o sistema deve impedir a inserção de mais caracteres e exibir um aviso sobre o limite.

Cenário 5 – Armazenamento correto da avaliação no banco de dados

- Dado que o usuário avaliou as duas respostas com notas e justificativas preenchidas,
- Quando o sistema processa essa avaliação,
- Então os dados devem ser armazenados corretamente no banco de dados e vinculados à pergunta avaliada.

6.1.2.4 US #05: Escolha da Melhor Resposta e Justificativa

Descrição:

Comentário PO:

– “Esta User Story é referente ao FrontEnd e BackEnd desta funcionalidade

Frontend: O usuário acessa a tela de "Escolha Final", onde ambas as respostas das LLMs são exibidas com seus critérios e notas. O usuário deve selecionar uma das respostas e preencher um campo de justificativa.

Backend: O sistema deve validar se a escolha do usuário é coerente com as notas atribuídas na avaliação. Caso haja incoerência, o sistema deve impedir a escolha e exibir uma mensagem ao usuário. Após a escolha correta e a justificativa, os dados devem ser armazenados no banco de dados.”

User Story:

“Como usuário, eu quero uma interface para poder escolher a melhor resposta entre as duas geradas pelas LLMs, para que o sistema registre minha decisão e justificativa.”

Critério de Aceitação:

FrontEnd:

- O sistema deve exibir ambas as respostas das LLMs, destacando seus critérios e notas atribuídas.
- O usuário deve poder selecionar apenas uma das respostas.
- Deve haver um campo de justificativa obrigatório para explicar a escolha (limite de 500 caracteres).
- O botão de "Confirmar Escolha" deve estar desabilitado até que a resposta seja escolhida e a justificativa seja preenchida.
- Se houver incoerência na escolha (ou seja, a resposta escolhida tiver uma nota inferior à outra), o sistema deve exibir um aviso e impedir a submissão.
- A interface deve seguir os padrões estabelecidos no Wireframe disponível no Figma: [LINK](#)

BackEnd:

- O sistema deve validar se a resposta escolhida é coerente com as notas atribuídas.
- Caso haja incoerência, o sistema deve impedir o usuário de confirmar a escolha e exibir um erro explicativo.
- Após a escolha coerente, o sistema deve preparar e salvar os dados no banco de dados, incluindo:
 - Pergunta original
 - Respostas das LLMs
 - Critérios com suas notas para cada Resposta
 - Resposta escolhida ao Final
 - Justificativa do usuário para a escolha Final

Cenários:**Cenário 1 – Exibição correta das respostas e notas**

- Dado que o usuário acessa a tela de "Escolha Final",
- Quando ambas as respostas das LLMs forem carregadas,
- Então o sistema deve exibir as respostas lado a lado, mostrando os critérios e suas respectivas notas.

Cenário 2 – Escolha de uma resposta e preenchimento da justificativa

- Dado que o usuário acessa a tela de "Escolha Final",
- Quando ele seleciona uma das respostas e digita uma justificativa válida,
- Então o botão "Confirmar Escolha" deve ser habilitado.

Cenário 3 – Tentativa de escolha sem justificativa

- Dado que o usuário acessa a tela de "Escolha Final",
- Quando ele seleciona uma resposta mas não preenche a justificativa,
- Então o sistema deve impedir a submissão e exibir um aviso de que a justificativa é obrigatória.

Cenário 4 – Escolha incoerente de resposta

- Dado que o usuário acessa a tela de "Escolha Final",
- Quando ele tenta escolher uma resposta com nota inferior à outra,

- Então o sistema deve impedir a submissão e exibir um erro informando que a escolha é incoerente.

Cenário 5 – Escolha válida e submissão bem-sucedida

- Dado que o usuário acessa a tela de "Escolha Final",
- Quando ele escolhe uma resposta coerente e preenche a justificativa,
- Então o sistema deve permitir a submissão e armazenar os dados no banco de dados.

Cenário 6 – Armazenamento correto no banco de dados

- Dado que o usuário confirmou a escolha de uma resposta,
- Quando o sistema processa os dados,
- Então a decisão do usuário deve ser armazenada corretamente, incluindo pergunta, respostas, critérios, justificativa e data/hora.

6.1.2.5 US #06: Armazenamento das Avaliações para Fine-Tuning

Descrição:

Comentário PO:

– “*Está User Story, tem foco no POST das informações de avaliação no banco de dados, garantindo que todas as informações necessárias sejam armazenadas corretamente. O objetivo é definir a estrutura do banco de dados e os dados que devem ser persistidos.*”

User Story:

“*Como Administrador, eu quero que as avaliações dos usuários sobre as LLMs sejam armazenadas em um BD, para que possam ser utilizadas em processos de fine-tuning futuramente.*”

Critério de Aceitação:

- O sistema deve receber os dados de avaliação dos usuários e armazená-los corretamente no banco de dados.
- Cada avaliação deve ser vinculada à respectiva pergunta e respostas avaliadas.
- O sistema deve garantir que nenhuma avaliação seja salva com dados incompletos ou inválidos.

- Caso ocorra um erro ao salvar no banco, o sistema deve registrar logs e retornar uma mensagem de erro ao serviço chamador.
- Os seguintes dados devem ser armazenados:
 - ID do usuário que realizou a avaliação
 - ID da pergunta (prompt)
 - O prompt que o usuário utilizou.
 - ID das respostas das LLMs avaliadas
 - Respostas 1 e 2 geradas pelas LLMs
 - Notas atribuídas a cada critério da Resposta 1
 - Notas atribuídas a cada critério da Resposta 2
 - ID da resposta selecionada no Final
 - Justificativas fornecidas pelo usuário

Cenários:

Cenário 1 – Armazenamento bem-sucedido da avaliação

- Dado que um usuário finaliza a avaliação de uma resposta,
- Quando ele submete os dados ao sistema,
- Então o sistema deve armazenar todas as informações corretamente no banco de dados.

Cenário 2 – Tentativa de armazenar avaliação incompleta

- Dado que o usuário submeteu uma avaliação,
- Quando a requisição não contém todas as informações obrigatórias (exemplo: falta de notas ou justificativa),
- Então o sistema deve rejeitar a requisição e exibir um erro informando quais dados estão ausentes.

Cenário 3 – Registro de erro ao tentar salvar no banco de dados

- Dado que o sistema recebe uma avaliação para armazenar,
- Quando ocorre um erro interno no banco de dados,
- Então o sistema deve registrar um log detalhado da falha e retornar uma mensagem de erro para o serviço chamador.

6.1.2.6 US #10: Administrador Pré-Cadastrado no Sistema

Descrição:

Comentário PO:

– “Essa User Story trata do mocking do usuário administrador no momento da inicialização do banco de dados, garantindo que o primeiro acesso ao sistema seja feito por um usuário pré-cadastrado com permissões administrativas.”

User Story:

“Como Administrador, eu gostaria de ser o primeiro usuário do sistema, já devidamente pré inserido no banco de dados, para que possa acessar a aplicação.”

Critério de Aceitação:

- O sistema deve criar um usuário Administrador padrão automaticamente ao inicializar o banco de dados pela primeira vez.
- O Administrador pré-cadastrado deve ter os seguintes atributos:
 - ID único
 - Nome padrão (exemplo: "Administrador Padrão")
 - E-mail fixo (exemplo: admin@domrock.com)
 - Senha inicial criptografada (exemplo: senha-gerada-automaticamente)
- O sistema deve garantir que não sejam criados múltiplos usuários administradores na inicialização, apenas um único Admin padrão.
- Caso o banco de dados já tenha um Administrador cadastrado, o sistema não deve recriar um novo.
- O sistema deve mostrar um aviso na inicialização informando se o Administrador foi criado ou se já existia um cadastro.

Cenários:

Cenário 1 – Criação automática do Administrador na inicialização do banco de dados

- Dado que o banco de dados é inicializado pela primeira vez,

- Quando o sistema verifica que não há usuários cadastrados,
- Então um usuário Administrador padrão deve ser criado automaticamente com credenciais predefinidas.

Cenário 2 – Garantia de que o Administrador não seja recriado se já existir

- Dado que o banco de dados já possui um Administrador cadastrado,
- Quando o sistema for reiniciado,
- Então o sistema não deve criar um novo Administrador, garantindo que não haja duplicatas.

Cenário 3 – Registro de logs na inicialização do banco

- Dado que o sistema inicia o banco de dados,
- Quando o Administrador for criado automaticamente ou detectado,
- Então um log deve ser gerado informando o status da criação.

6.2 SPRINT 02

Rank	Prioridade	User Story	Story Points	Sprint	Requisito do Cliente	Status
7	Alta	Como usuário, eu quero ser alertado caso minha escolha entre LLM1 e LLM2 não esteja coerente com minha avaliação, para que eu possa revisar minha decisão antes de finalizar.	3	2	R04	✓
8	Alta	Como usuário, eu quero que as respostas das LLMs sejam enriquecidas com informações relevantes da base de dados (vetorizada), para que sejam mais precisas.	5	2	R07	✓
15	Média	Como usuário, eu quero poder revisar minha escolha antes de submetê-la, para que eu tenha certeza de que minha decisão está correta.	2	2	R04	✓
19	Baixa	Como usuário, eu quero ser informado com mensagens de erro caso ocorra demora excessiva no envio do prompt ou na resposta das LLMs, ou outros erros, para que eu possa entender o problema e tentar novamente.	2	2	R03	✓
21	Baixa	Como usuário, eu quero poder receber	2	2	R03	✓

		mensagens claras sobre o status das avaliações, para ter certeza de que minha avaliação foi registrada corretamente				
22	Baixa	Como usuário, eu quero poder voltar para telas anteriores durante o processo de avaliação, para que eu possa corrigir informações antes de enviar a decisão final.	1	2	R03/R04	✓

6.2.1 DoD & DoR – Sprint 2

6.2.1.1 Definition Of Ready (DOR) – Sprint 2

Critério	Descrição
Clareza na Descrição	A User Story está escrita no formato “Como [persona], quero [ação] para que [objetivo]”.
Critérios de Aceitação Definidos	A história possui critérios objetivos que indicam o que é necessário para considerá-la concluída.
Cenários de Teste Especificados	A história tem pelo menos 1 cenário de teste estruturado (Dado, Quando, Então).
Independente	A história pode ser implementada sem depender de outra tarefa da mesma Sprint.
Escopo Técnico Validado	Há clareza se a história envolve frontend, backend ou ambos — com pontos de integração definidos.
Estimável	A história foi pontuada no Planning Poker ou tem uma estimativa clara.
Referências ou Documentos de Apoio	Onde necessário, foram fornecidos arquivos, links (ex: base vetorizada), ou instruções adicionais.
Validação com PO e equipe	A história foi discutida em refinamento ou planning e validada com o time técnico.
Alinhamento com arquitetura atual	A funcionalidade proposta está coerente com o funcionamento já entregue na Sprint 1.

6.2.1.2 Definition Of Done (DOD) – Sprint 2

Critério	Descrição
Critérios de Aceitação atendidos	Todos os critérios definidos na US foram implementados e validados com sucesso.
Cenários de Teste executados e aprovados	Todos os cenários descritos foram validados manualmente ou por teste automatizado.
Dados persistidos corretamente	Informações sensíveis como prompt, resposta, avaliação ou contexto vetorizado foram salvos conforme esperado.
Feedback visual implementado	Funcionalidades como pop-ups, mensagens de erro ou barras de progresso estão claras e acessíveis ao usuário.
Fluxo seguro e controlado	Não há caminhos quebrados nem submissões incoerentes no fluxo de avaliação ou navegação.
Código revisado (Code Review)	A US passou por revisão entre pares antes da integração.
Integrado com o restante da aplicação	A funcionalidade foi testada junto com o fluxo completo do sistema (Ex: Envio → Resposta → Avaliação → Escolha).
Documentação Técnica atualizada	Se aplicável, a lógica de RAG, manipulação da base vetorizada ou instruções de uso estão registradas.
Validação final com o PO	O PO testou e confirmou que a funcionalidade atende ao esperado.
Pronta para Deploy	A funcionalidade pode ser entregue ao ambiente de produção/testes finais sem pendências.

6.2.2 Cenários E Critérios De Aceite – User Stories – Sprint 2

6.2.2.1 US #07: Alerta de Inconsistência na Escolha da Resposta

Descrição:

Comentário PO:

– “A lógica de coerência já está implementada, impedindo que o usuário selecione uma resposta incoerente com base nas notas atribuídas. No entanto, o sistema apenas bloqueia a interação com o cursor sendo travado.

O objetivo agora desta User Story é fornecer feedback visual e explicativo ao usuário por meio de um Pop-up ou Modal, informando por que a escolha não é permitida.”

User Story:

"Como usuário, eu quero ser alertado caso minha escolha entre LLM1 e LLM2 não esteja coerente com minha avaliação, para que eu possa revisar minha decisão antes de finalizar."

Critério de Aceitação:

- O sistema deve identificar automaticamente quando a resposta escolhida pelo usuário é incoerente com os critérios avaliados.
- Quando o usuário tentar selecionar uma resposta incoerente, deve ser exibido um modal explicativo.
- O modal deve informar claramente que a resposta escolhida não possui a maior pontuação média entre os critérios avaliados.
- O modal deve conter um botão de "Entendi", que fecha a mensagem e mantém a seleção inalterada.
- O sistema não deve permitir a submissão da escolha incoerente.

Cenários:

Cenário 1 – Seleção incoerente com mensagem explicativa

- Seleção incoerente com mensagem explicativa
- Dado que o usuário finalizou a avaliação individual das duas respostas,
- Quando ele tenta selecionar como melhor resposta aquela que obteve menor pontuação,

- Então o sistema deve exibir um modal explicando que a escolha não é coerente com as avaliações e, por isso, não será permitida.

Cenário 2 – Modal de explicação com opção de fechar

- Dado que o modal de inconsistência foi exibido,
- Quando o usuário clicar no botão "Entendi",
- Então o modal deve ser fechado até que o usuário clique novamente em uma resposta incoerente.

Cenário 3 – Tentativa de envio bloqueada em caso de inconsistência

- Dado que o usuário tenta confirmar uma resposta com pontuação inferior,
- Quando a inconsistência for detectada,
- Então o sistema deve exibir o alerta e não permitir o envio da escolha até que uma resposta coerente seja selecionada.

Cenário 4 – Comportamento ao selecionar resposta coerente

- Dado que o usuário seleciona uma resposta coerente (com maior pontuação),
- Quando ele confirmar a escolha,
- Então o sistema deve permitir a submissão normalmente, sem exibir o modal.

6.2.2.2 US #08: Enriquecimento das Respostas com Base Vetorizada - RAG Parte 1

Descrição:

Comentário PO:

– “Esta funcionalidade é sobre a introdução da técnica de RAG (Retrieval-Augmented Generation) ao fluxo atual da aplicação. A base de dados fornecida pelo cliente (sobre Alzheimer) já se encontra vetorizada e pronta para uso via ChromaDB. O enriquecimento será feito através da recuperação semântica de trechos relevantes que serão incluídos ao prompt original antes do envio às LLMs.”

User Story:

"Como usuário, eu quero que as respostas das LLMs sejam enriquecidas com informações relevantes da base de dados (vetorizada), para que sejam mais precisas."

Critério de Aceitação:

- O sistema deve consultar a base vetorizada antes de enviar o prompt à LLM.
- O contexto retornado da base deve ser incluído no prompt de forma visível e rastreável.
- A resposta da LLM deve incorporar elementos da base vetorizada.
- O processo de recuperação e enriquecimento deve ser transparente ao usuário (mas pode ser visualizado tecnicamente pela equipe).
- A funcionalidade deve ser aplicável tanto à LLM 1 quanto à LLM 2.
- A base utilizada deve ser a chroma.sqlite3 entregue pela Dom Rock. [LINK](#)
- A resposta enriquecida deve ser persistida com o prompt original e o contexto utilizado.

Cenários:

Cenário 1 – Resposta enriquecida com contexto encontrado

- Dado que o usuário envia um prompt sobre Alzheimer
- Quando existe conteúdo relacionado na base vetorizada
- Então o sistema deve recuperar esse conteúdo e incluí-lo no prompt enviado à LLM
- E a resposta deve conter menções ou dados provenientes do contexto retornado

Cenário 2 – Resposta sem contexto aplicável

- Dado que o usuário envia um prompt para o qual não há correspondência semântica relevante na base
- Quando o sistema tenta recuperar contexto
- Então o prompt deve ser enviado sem enriquecimento adicional
- E a aplicação deve registrar que nenhum contexto foi utilizado

Cenário 3 – Validação do contexto utilizado

- Dado que a LLM retornou uma resposta enriquecida
- Quando o desenvolvedor ou tester analisa os dados persistidos
- Então o contexto utilizado deve estar registrado junto ao prompt e à resposta final

6.2.2.3 US #15: *Revisão da Escolha Antes da Submissão*

Descrição:

Comentário PO:

– “Nesta User Story o cliente sugeriu que o sistema ofereça uma etapa de confirmação da escolha, onde o usuário possa visualizar qual resposta foi selecionada como final. Se ele confirmar, o sistema deve persistir a decisão no banco de dados. Caso não confirme, ele deve poder navegar livremente pelo sistema e corrigir a escolha antes de submeter.”

User Story:

"Como usuário, eu quero poder revisar minha escolha antes de submetê-la, para que eu tenha certeza de que minha decisão está correta."

Critério de Aceitação:

- O sistema deve apresentar uma tela de revisão após a seleção da melhor resposta.
- A tela/modal de revisão deve exibir a resposta escolhida, os critérios avaliados e a justificativa fornecida.
- Deve haver dois botões:
 - "Confirmar Escolha": para persistir a decisão no banco de dados.
 - "Voltar e Editar": para retornar à etapa anterior e modificar a escolha.
- A escolha só deve ser persistida no banco de dados após a confirmação explícita do usuário.
- A tela/modal de revisão deve estar disponível somente após o preenchimento completo dos critérios e justificativa.
- O sistema deve manter os dados preenchidos ao retornar para a etapa anterior.

Cenários:

Cenário 1 – Exibição da tela de revisão após escolha final

- Dado que o usuário selecionou a melhor resposta e preencheu a justificativa,
- Quando ele avança para a próxima etapa,
- Então o sistema deve exibir uma “tela de revisão”(ou Modal) com os dados da escolha e os critérios avaliados.

Cenário 2 – Confirmação da escolha e persistência no banco

- Dado que o usuário está na tela de revisão (ou Modal),
- Quando ele clica em "Confirmar Escolha",
- Então o sistema deve persistir os dados no banco de dados e redirecionar o usuário para a tela de encerramento/feedback.

Cenário 3 – Voltar e editar a escolha antes da submissão

- Dado que o usuário está na tela de revisão (ou Modal),
- Quando ele clica em "Voltar e Editar",
- Então o sistema deve levá-lo de volta à tela de escolha final e manter os dados preenchidos anteriormente.

Cenário 4 – Tentativa de acessar a tela de revisão com dados incompletos

- Dado que o usuário ainda não preencheu completamente os critérios ou justificativa,
- Quando ele tenta avançar para a tela de revisão,
- Então o sistema deve exibir uma mensagem alertando que a escolha só pode ser revisada após o preenchimento completo.

Cenário 5 – Garantia de que a submissão só ocorre após confirmação

- Dado que o usuário acessa a tela de revisão,
- Quando ele não clica em "Confirmar Escolha",
- Então o sistema não deve salvar nenhuma informação no banco de dados até que a confirmação seja feita.

6.2.2.4 US #19: Alerta de Timeout no Envio ou Resposta das LLMs

Descrição:

Comentário PO:

– “Esta User Story unifica as anteriores relacionadas ao envio e resposta das LLMs, (A6S-26: Alerta de Erro na Obtenção das Respostas) e (A6S-115: Alerta de Erro no Envio do Prompt).

O objetivo é informar o usuário em caso de demora (timeout) na comunicação com as LLMs, seja no envio do prompt ou na obtenção das respostas e tratar outros erros que prejudiquem a experiência do usuário. A mensagem deve ser clara e incluir um botão para reenviar o prompt. O sistema deve identificar falhas por timeout e tratá-las de forma uniforme para o usuário.”

User Story:

"Como usuário, eu quero ser informado com mensagens de erro caso ocorra demora excessiva no envio do prompt ou na resposta das LLMs, ou outros erros, para que eu possa entender o problema e tentar novamente. "

Critério de Aceitação:

- O sistema deve detectar timeouts tanto no envio quanto na resposta das LLMs.
- Caso o tempo de resposta exceda o limite definido, deve ser exibida uma mensagem de erro amigável ao usuário.
- A mensagem deve conter um botão "Tentar novamente", que reenvia o prompt sem que o usuário precise reescrevê-lo.
- O sistema deve impedir o usuário de seguir com o fluxo de avaliação até obter uma resposta válida.

Cenários:**Cenário 1 – Timeout durante envio do prompt**

- Dado que o usuário digitou um prompt válido,
- Quando o sistema não consegue enviar o prompt dentro do tempo limite,
- Então uma mensagem de erro deve ser exibida informando a falha no envio, com botão para reenviar.

Cenário 2 – Timeout na resposta das LLMs

- Dado que o prompt foi enviado com sucesso,
- Quando a resposta de uma ou ambas as LLMs demora além do tempo configurado,
- Então o sistema deve exibir uma mensagem de erro explicando a falha na resposta, com botão de reenvio.

Cenário 3 – Reenvio após timeout bem-sucedido

- Dado que o usuário recebeu uma mensagem de timeout,
- Quando ele clica em "Tentar novamente",
- Então o sistema deve reenviar o prompt e, se bem-sucedido, seguir o fluxo normalmente.

6.2.2.5 US #21: *Confirmação que a Avaliação foi Registrada - Status***Descrição:**

Comentário PO:

– “Essa User Story não se resume a uma simples confirmação (ex: “avaliação salva com sucesso”), mas busca aprimorar a UX, através de feedbacks visuais contínuos (como uma “Progress Bar”, status ou indicações visuais). A proposta é que o sistema mostre em tempo real o progresso do processo de avaliação, reforçando a confiança do usuário sobre a ação que está realizando e ajudando a orientá-lo nas próximas etapas.”

User Story:

"Como usuário, eu quero poder receber mensagens claras sobre o status das avaliações, para ter certeza de que minha avaliação foi registrada corretamente"

Critério de Aceitação:

- O sistema deve exibir feedback visual claro de progresso nas etapas da avaliação (ex: avaliação da LLM1, LLM2, escolha final).
- Após a submissão de uma avaliação individual, o sistema deve mostrar uma mensagem de sucesso (ex: "Avaliação registrada com sucesso").
- O usuário deve conseguir visualizar em qual etapa da jornada de avaliação ele está.
- A progressão deve ser atualizada automaticamente conforme o usuário completa cada etapa.
- O sistema deve impedir que o usuário avance para uma próxima etapa sem concluir a atual.

Cenários:

Cenário 1 – Exibição da etapa atual na progress bar

- Dado que o usuário iniciou o processo de avaliação,
- Quando ele estiver avaliando a resposta da LLM1,
- Então o sistema deve exibir visualmente a indicação "Etapa 1 de 3 – Avaliando Resposta da LLM1".

Cenário 2 – Falha na comunicação com ambas as LLMs

- Dado que o usuário concluiu a avaliação de uma resposta,
- Quando ele clicar em "Confirmar Avaliação",
- Então o sistema deve exibir uma mensagem clara de que a avaliação foi registrada com sucesso.

Cenário 3 – Atualização da progress bar após cada etapa

- Dado que o usuário está na etapa de avaliação,
- Quando ele concluir a avaliação da LLM1 e for para a LLM2,

- Então a progress bar deve ser atualizada automaticamente para refletir "Etapa 2 de 3".prompt.

Cenário 4 – Bloqueio de avanço sem concluir a etapa atual

- Dado que o usuário está na etapa de avaliação da LLM1,
- Quando ele tenta ir para a próxima etapa sem concluir os critérios obrigatórios,
- Então o sistema deve impedir o avanço e alertar sobre a necessidade de completar a avaliação atual.

6.2.2.6 US #22: Navegação para Telas Anteriores Durante a Avaliação

Descrição:

Comentário PO:

– “A funcionalidade foi entregue durante a Sprint 1, como consequência do fluxo de navegação implementado. No entanto, é importante verificar se o comportamento está adequado ao esperado, garantindo que os dados preenchidos sejam mantidos ao retornar e que o usuário não se perca no processo de avaliação. Caso identificado algum ponto de melhoria, ajustes poderão ser feitos como refinamento.

Esta User Story foi unida com a US: (A6S-29: Preservação do Progresso ao Retornar para Telas Anteriores), após avaliação minha, PO @Ivan Germano Rosendo de Oliveira Duarte, ao verificar que, além de as duas User Stories estarem praticamente completas, também estão intrinsecamente unidas em funcionalidade, e por isso não tem necessidade de serem separadas.”

User Story:

"Como usuário, eu quero poder voltar para telas anteriores durante o processo de avaliação, para que eu possa corrigir informações antes de enviar a decisão final."

Critério de Aceitação:

- O usuário deve poder voltar livremente para telas anteriores durante o processo de avaliação (ex: da escolha final para avaliação individual).
- A navegação reversa não deve quebrar a lógica do fluxo, nem permitir submissão de dados parciais ou inconsistentes.

- Ao retornar, os dados previamente preenchidos devem ser mantidos (ex: notas, justificativas, seleção da melhor resposta).
- A interface deve oferecer botões claros para voltar à etapa anterior.
- O sistema deve garantir que, ao editar uma etapa anterior, o usuário possa prosseguir novamente sem perder progresso.

Cenários:

Cenário 1 – Retorno para etapa anterior com dados preservados

- Dado que o usuário está na tela de "Escolha Final",
- Quando ele clica para voltar à avaliação da LLM2,
- Então o sistema deve exibir os dados previamente preenchidos para essa resposta (notas e justificativas).

Cenário 2 – Retorno múltiplo sem quebra do fluxo

- Dado que o usuário volta da "Escolha Final" até a avaliação da LLM1,
- Quando ele modifica um critério e segue novamente até a tela final,
- Então o sistema deve permitir que ele continue normalmente com o fluxo de avaliação.

Cenário 3 – Navegação clara com botões visíveis

- Dado que o usuário está em qualquer etapa do processo,
- Quando ele desejar retornar,
- Então o botão de "Voltar" deve estar visível, acessível e funcional.

Cenário 4 – Evitar submissão com dados incompletos ao retornar

- Dado que o usuário altera informações em uma etapa anterior,
- Quando ele tenta enviar a avaliação final,
- Então o sistema deve validar novamente todos os critérios obrigatórios antes da submissão.

6.3 SPRINT 03

Rank	Prioridade	User Story	Story Points	Sprint	Requisito do Cliente	Status
11	Média	Como Administrador, eu quero cadastrar novos usuários na plataforma, para que somente pessoas autorizadas possam acessá-la.	17	3	R06	✓
12	Média	Como usuário, eu quero acessar a aplicação através de uma interface de login, para que somente usuários autorizados possam utilizar o sistema.	13	3	R06	✓
13	Média	Como usuário autorizado, eu quero acessar o sistema através de um login, para utilizar a aplicação.	23	3	R06	✓
14	Média	Como usuário autenticado, eu quero poder fazer o logout da aplicação de forma segura, para que meus dados não fiquem acessíveis a terceiros.	18	3	R06	✓
16	Média	Como Administrador, eu quero visualizar a lista de usuários cadastrados, para que eu possa gerenciar quem tem acesso ao sistema.	20	3	R06	✓
17	Média	Como Administrador, eu quero redefinir a senha de um usuário, para que eu possa ajudá-lo caso ele não consiga acessar a conta.	18	3	R06	✓
18	Média	Como Administrador, eu quero excluir usuários do sistema, para que possa revogar o acesso de usuários a aplicação.	13	3	R06	✓
24	Baixa	Como usuário, eu quero editar meus dados pessoais, para que eu possa manter minhas informações atualizadas.	16	3	R06	✓

6.3.1 DoD & DoR – Sprint 3

6.3.1.1 Definition Of Ready (DOR) – Sprint 3

Critério	Descrição
Clareza na Descrição	A User Story está escrita no formato “Como [persona], quero [ação] para que [objetivo]”.
Critérios de Aceitação Definidos	A história possui critérios objetivos que indicam o que é necessário para considerá-la concluída.
Cenários de Teste Especificados	A história tem pelo menos 1 cenário de teste estruturado (Dado, Quando, Então).
Independente	A história pode ser implementada sem depender de outra tarefa da mesma Sprint.
Referência Visual no Figma	A User Story que envolver front-end ou refatoração de layout, o protótipo correspondente no Figma deve estar disponível e vinculado à US.
Escopo Técnico Validado	Há clareza se a história envolve frontend, backend ou ambos — com pontos de integração definidos.
Perfil de Acesso Definido	Está claro se a US é acessível por usuário comum, administrador ou ambos.
Integração com Histórias Anteriores está Clara	A funcionalidade se conecta logicamente ao que já foi entregue nas Sprints 1 e 2.
Estimável	A história foi pontuada no Planning Poker ou tem uma estimativa clara.
Referências ou Documentos de Apoio	Onde necessário, foram fornecidos arquivos, links (ex: base vetorizada), ou instruções adicionais.
Validação com PO e equipe	A história foi discutida em refinamento ou planning e validada com o time técnico.

6.3.1.2 Definition Of Done (DOD) – Sprint 3

Critério	Descrição
Critérios de Aceitação Atendidos	Todos os critérios definidos na US foram implementados e validados com sucesso.
Regras de Segurança Aplicadas	Itens como controle de acesso, logout, edição e redefinição de senha atendem aos requisitos de segurança.
Dados Persistidos Corretamente	As alterações (cadastro, edição, exclusão) são refletidas corretamente no banco de dados.
Comportamento por Perfil Validado	Testes foram feitos tanto para usuários comuns quanto para administradores.
Integração com o Sistema Existente Verificada	A funcionalidade não quebra o fluxo atual de uso (ex: login → avaliação → edição de perfil).
Código revisado (Code Review)	A US passou por revisão entre pares antes da integração.
Integrado com o restante da aplicação	A funcionalidade foi testada junto com o fluxo completo do sistema (Ex: Envio → Resposta → Avaliação → Escolha).
Código Revisado e Validado	A funcionalidade foi submetida a code review e validada por pelo menos outro membro da equipe.
Documentação Atualizada	Foram registrados endpoints, lógicas de autenticação, permissões e observações técnicas
PO Validou Funcionalmente	O PO testou e aprovou a funcionalidade em ambiente de homologação ou desenvolvimento.

6.3.2 Cenários E Critérios De Aceite – User Stories – Sprint 2

6.3.2.1 US #11: Cadastro de Novos Usuários na Plataforma

Descrição:

Comentário PO:

– “Esta funcionalidade compõe o módulo de controle de acesso, permitindo ao perfil de Administrador incluir usuários na base do sistema. O cadastro deve contemplar dados essenciais como nome, e-mail, tipo de perfil (comum ou administrador) e senha inicial. O objetivo é garantir que somente pessoas autorizadas tenham acesso ao sistema, conforme escopo validado com o cliente.”

User Story:

"Como Administrador, eu quero cadastrar novos usuários na plataforma, para que somente pessoas autorizadas possam acessá-la."

Critério de Aceitação:

- A funcionalidade deve estar disponível apenas para usuários com perfil de Administrador.
- O formulário de cadastro deve conter, no mínimo: nome, e-mail, senha inicial e tipo de usuário.
- O sistema deve validar se o e-mail já está em uso antes de permitir o cadastro.
- Após o cadastro, o novo usuário deve estar ativo e apto a fazer login.
- O sistema deve exibir uma mensagem de confirmação após o cadastro bem-sucedido.
- As informações do novo usuário devem ser persistidas no banco de dados.
- Deve ser possível acessar a lista de usuários e visualizar o novo cadastro.

Cenários:

Cenário 1 – Cadastro bem-sucedido

- Dado que o Administrador acessa a tela de cadastro de usuários,
- Quando ele preenche os campos obrigatórios com dados válidos,
- Então o sistema deve salvar o novo usuário e exibir uma mensagem de sucesso.

Cenário 2 – Tentativa de cadastro com e-mail duplicado

- Dado que o Administrador tenta cadastrar um novo usuário,

- Quando o e-mail informado já estiver registrado no sistema,
- Então o sistema deve impedir o cadastro e exibir uma mensagem de erro.

Cenário 3 – Acesso restrito à funcionalidade de cadastro

- Dado que um usuário comum está logado no sistema,
- Quando ele tenta acessar a tela de cadastro de usuários,
- Então o sistema deve bloquear o acesso e redirecioná-lo ou exibir uma mensagem de permissão negada.

Cenário 4 – Visualização do novo usuário na lista

- Dado que o Administrador cadastrou um novo usuário,
- Quando ele acessa a lista de usuários,
- Então o novo usuário deve aparecer com as informações inseridas e status "ativo".

6.3.2.2 US #12: Interface de Login para Acesso ao Sistema

Descrição:

Comentário PO:

– “Esta funcionalidade representa a porta de entrada do sistema. A tela de login deve ser clara, responsiva e segura, permitindo que usuários previamente cadastrados façam autenticação utilizando seu e-mail e senha. A autenticação será validada contra o banco de dados, garantindo que apenas usuários ativos possam acessar a aplicação.”

User Story:

"Como usuário, eu quero acessar a aplicação através de uma interface de login, para que somente usuários autorizados possam utilizar o sistema."

Critério de Aceitação:

- A interface deve conter campos para e-mail e senha.
- Deve existir um botão de "Entrar" que só é habilitado após o preenchimento dos campos.
- O sistema deve validar se o e-mail e senha correspondem a um usuário ativo.
- Caso os dados estejam corretos, o usuário deve ser redirecionado à área interna da aplicação

- Caso os dados estejam incorretos, o sistema deve exibir uma mensagem de erro clara e não técnica.

Cenários:

Cenário 1 – Login bem-sucedido

- Dado que o usuário possui um cadastro ativo,
- Quando ele preenche e-mail e senha corretamente e clica em "Entrar",
- Então o sistema deve autenticar o usuário e redirecioná-lo para a área interna.

Cenário 2 – Tentativa com credenciais incorretas

- Dado que o usuário tenta fazer login,
- Quando ele insere um e-mail ou senha inválidos,
- Então o sistema deve exibir uma mensagem de erro como "Usuário ou senha inválidos" sem detalhar o que está errado.

Cenário 3 – Tentativa com usuário inativo

- Dado que um usuário está cadastrado mas inativo,
- Quando ele tenta fazer login,
- Então o sistema deve exibir uma mensagem informando que o usuário está desabilitado.

6.3.2.3 US #13: Autenticação de Usuários Autorizados

Descrição:

Comentário PO:

– “Esta User Story complementa a (#US12: Interface de Login para Acesso ao Sistema) , focando na validação de permissões e no comportamento após a autenticação. Aqui o objetivo é garantir que somente usuários com credenciais válidas e permissões apropriadas possam acessar a aplicação e serem direcionados à área correta do sistema (por exemplo: área de usuário comum ou painel de administrador).”

User Story:

"Como usuário autorizado, eu quero acessar o sistema através de um login, para utilizar a aplicação."

Critério de Aceitação:

- O sistema deve verificar se o usuário está autorizado e ativo no momento do login.
- Após o login, o sistema deve direcionar o usuário à interface adequada conforme seu perfil (comum ou administrador).
- Se o usuário tentar acessar rotas não permitidas ao seu perfil, o sistema deve bloquear o acesso.
- O sistema deve validar o token em cada requisição autenticada.
- Sessões devem expirar após tempo definido ou logout.

Cenários:

Cenário 1 – Acesso autorizado com redirecionamento correto

- Dado que o usuário está ativo e com login válido,
- Quando ele realiza o login com sucesso,
- Então o sistema deve redirecioná-lo para a interface correspondente ao seu perfil.

Cenário 2 – Tentativa de acesso a área não autorizada

- Dado que o usuário comum está logado,
- Quando ele tenta acessar uma rota exclusiva de administrador,
- Então o sistema deve negar o acesso e exibir uma mensagem de permissão negada.

Cenário 3 – Token inválido ou expirado

- Dado que o usuário está logado,
- Quando o token de autenticação expira ou é corrompido,
- Então o sistema deve forçar logout e redirecionar o usuário para a tela de login.

Cenário 4 – Armazenamento e uso do token

- Dado que o login foi realizado com sucesso,
- Quando o sistema precisar fazer chamadas autenticadas,
- Então o token de autenticação deve ser incluído nas requisições de forma segura.

Cenário 5 – Acesso negado para usuário inativo ou removido

- Dado que o usuário teve seu acesso revogado,
- Quando ele tenta realizar login,
- Então o sistema deve barrar o acesso com mensagem informando que o usuário está.

6.3.2.4 US #14: Logout Seguro da Aplicação

Descrição:

Comentário PO:

– “Esta funcionalidade garante a finalização segura da sessão do usuário, removendo os dados de autenticação e redirecionando para a tela de login. É essencial para preservar a segurança dos dados pessoais e profissionais, principalmente em ambientes compartilhados ou públicos.”

User Story:

"Como usuário autenticado, eu quero poder fazer o logout da aplicação de forma segura, para que meus dados não fiquem acessíveis a terceiros."

Critério de Aceitação:

- O sistema deve disponibilizar um botão visível de "Logout" na interface após o login.
- Ao clicar em logout, o token de autenticação deve ser removido (ex: do localStorage ou cookies).
- O sistema deve redirecionar o usuário para a tela de login imediatamente após o logout.
- O sistema não deve permitir acesso a rotas protegidas após logout, mesmo via URL.
- Caso o usuário tente retornar via botão "Refresh", ele deve ser redirecionado novamente para o login.

Cenários:

Cenário 1 – Logout bem-sucedido com redirecionamento

- Dado que o usuário está autenticado,
- Quando ele clica no botão de logout,
- Então o sistema deve apagar o token e redirecionar o usuário para a tela de login.

Cenário 2 – Bloqueio de rotas após logout

- Dado que o usuário realizou logout,
- Quando ele tenta acessar uma URL interna da aplicação,
- Então o sistema deve barrar o acesso e forçar retorno à tela de login.

Cenário 3 – Botão "Refresh" do navegador

- Dado que o usuário fez logout,
- Quando ele pressiona o botão de voltar do navegador,

- Então ele não deve conseguir acessar páginas protegidas e deve ser redirecionado para o login.

Cenário 4 – Validação da remoção de dados de sessão

- Dado que o usuário clicou em logout,
- Quando o desenvolvedor ou QA inspeciona o navegador,
- Então o token de autenticação e qualquer dado de sessão devem estar removidos.

6.3.2.5 US #16: Visualização e Gestão da Lista de Usuários

Descrição:

Comentário PO:

– “Essa funcionalidade é parte essencial do painel administrativo. Permite que o administrador visualize todos os usuários cadastrados na plataforma com informações essenciais como nome, e-mail, status (ativo/inativo) e tipo de perfil (usuário comum ou administrador). Essa tela será base para ações futuras como edição, redefinição de senha ou exclusão de usuários.”

User Story:

"Como Administrador, eu quero visualizar a lista de usuários cadastrados, para que eu possa gerenciar quem tem acesso ao sistema."

Critério de Aceitação:

- A funcionalidade deve ser exclusiva para administradores autenticados.
- A lista deve exibir nome, e-mail, tipo de usuário e status de cada usuário.
- O sistema deve suportar paginação ou scroll infinito, se houver muitos registros.
- O administrador deve poder pesquisar e filtrar usuários por nome ou email.
- Cada linha da lista deve conter ações acessíveis como editar, redefinir senha, excluir]]
- A tela deve seguir o padrão definido no figma em desktop.

Cenários:

Cenário 1 – Acesso autorizado à lista de usuários

- Dado que o administrador está autenticado,
- Quando ele acessa o painel de usuários,
- Então o sistema deve exibir a lista completa com as informações de cada usuário.

Cenário 2 – Acesso bloqueado para usuários comuns

- Dado que um usuário comum está autenticado,
- Quando ele tenta acessar a rota da lista de usuários,
- Então o sistema deve exibir uma mensagem de permissão negada ou redirecionar para a página principal.

Cenário 3 – Filtro e pesquisa de usuários

- Dado que o administrador está na lista de usuários,
- Quando ele pesquisa pelo nome ou e-mail de um usuário,
- Então a lista deve ser atualizada com os resultados correspondentes.

Cenário 4 – Visualização de status e tipo de perfil

- Dado que o administrador acessa a lista de usuários,
- Quando ele analisa a tabela,
- Então deve ser possível ver claramente se o usuário está ativo ou inativo, e seu perfil (comum ou admin).

6.3.2.6 US #17: Redefinição de Senha para Usuários como Administrador

Descrição:

Comentário PO:

– “Essa funcionalidade é fundamental para a manutenção do suporte administrativo. Permite que o administrador gere uma nova senha temporária para usuários que perderam o acesso, sem necessidade de intervenção técnica. Estamos partindo do pressuposto que usuários terão contato direto com o Adm através de alguma plataforma de comunicação. Nossa aplicação não contempla isso.”

User Story:

"Como Administrador, eu quero redefinir a senha de um usuário, para que eu possa ajudá-lo caso ele não consiga acessar a conta."

Critério de Aceitação:

- Ao selecionar a ação, o sistema deve gerar uma nova senha temporária segura.

- A nova senha deve ser exibida ao administrador, para o mesmo tomar as medidas necessárias para enviar ao usuário.
- A senha antiga deve ser invalidada imediatamente após a redefinição.

Cenários:

Cenário 1 – Redefinição de senha por administrador

- Dado que o administrador está autenticado e acessa a lista de usuários,
- Quando ele seleciona a opção de redefinir a senha de um usuário,
- Então o sistema deve gerar uma nova senha temporária e invalidar a senha antiga.

Cenário 2 – Persistência e segurança da nova senha

- Dado que a senha foi redefinida,
- Quando o desenvolvedor ou QA inspeciona o banco de dados,
- Então a nova senha deve estar criptografada e a senha antiga não deve mais estar ativa.

6.3.2.7 US #18: Exclusão de Usuários pelo Administrador

Descrição:

Comentário PO:

– “Essa funcionalidade permite que o administrador revogue permanentemente o acesso de usuários, garantindo que contas desnecessárias, inativas ou indevidas não permaneçam no sistema. A exclusão deve ser segura e irreversível, ou alternativamente, marcada como “inativo”, caso opte-se por exclusão lógica.”

User Story:

"Como Administrador, eu quero excluir usuários do sistema, para que possa revogar o acesso de usuários a aplicação."

Critério de Aceitação:

- A opção de excluir usuários deve estar disponível apenas para administradores autenticados.
- O sistema deve exibir uma confirmação antes da exclusão para evitar exclusões acidentais.
- O usuário excluído não deve mais conseguir acessar o sistema.
- A exclusão deve remover ou desativar o registro do usuário no banco de dados.

- O sistema não deve permitir que o administrador exclua sua própria conta.

Cenários:

Cenário 1 – Exclusão bem-sucedida de um usuário

- Dado que o administrador está na lista de usuários,
- Quando ele seleciona a opção de excluir um usuário e confirma a ação,
- Então o sistema deve remover ou desativar o usuário e exibir uma mensagem de sucesso.

Cenário 2 – Confirmação antes da exclusão

- Dado que o administrador clica em "Excluir",
- Quando o sistema exibe uma janela de confirmação,
- Então a exclusão só deve ocorrer se o administrador confirmar a ação.

Cenário 3 – Tentativa de login por usuário excluído

- Dado que o usuário foi excluído,
- Quando ele tenta fazer login,
- Então o sistema deve negar o acesso e informar que o usuário não existe ou foi removido.

Cenário 4 – Tentativa de exclusão da própria conta pelo admin

- Dado que o administrador está autenticado,
- Quando ele tenta excluir sua própria conta,
- Então o sistema deve impedir a ação e exibir uma mensagem de bloqueio.

6.3.2.8 US #24: *Edição de Dados Pessoais*

Descrição:

Comentário PO:

– “Essa funcionalidade dá ao usuário comum a capacidade de editar suas próprias informações, como nome e senha. O e-mail pode ser mantido como campo não editável. A edição deve ser segura, validada e refletida imediatamente na interface e no banco de dados.”

User Story:

"Como usuário, eu quero editar meus dados pessoais, para que eu possa manter minhas informações atualizadas."

Critério de Aceitação:

- O usuário deve acessar uma tela ou modal de "Minha Conta" com suas informações atuais preenchidas.
- O usuário deve poder editar seu nome e senha.
- O campo de e-mail deve ser exibido, mas não deve ser editável.
- O sistema deve validar os campos antes de permitir a atualização.
- Após salvar, uma mensagem de confirmação deve ser exibida.
- As alterações devem ser persistidas no banco de dados com segurança, nova senha criptografada.
- A senha deve seguir critérios mínimos (ex: 8 caracteres, ao menos uma letra e um número).

Cenários:

Cenário 1 – Edição bem-sucedida de dados pessoais

- Dado que o usuário está autenticado,
- Quando ele acessa a tela de edição, altera seu nome e salva,
- Então o sistema deve atualizar os dados e exibir uma mensagem de sucesso.

Cenário 2 – Alteração de senha com validação

- Dado que o usuário deseja alterar sua senha,
- Quando ele insere uma nova senha válida e a confirma,
- Então o sistema deve atualizar a senha no banco e aplicar a criptografia.

Cenário 3 – Tentativa de edição com campos inválidos

- Dado que o usuário acessa a tela de edição,
- Quando ele tenta salvar sem preencher todos os campos obrigatórios ou com dados inválidos,
- Então o sistema deve exibir mensagens de erro apropriadas e bloquear o envio.

Cenário 4 – Impossibilidade de alterar e-mail

- Dado que o usuário acessa a tela de edição,
- Quando ele tenta alterar o campo de e-mail,
- Então o campo deve estar desabilitado para edição.

Cenário 5 – Validação da segurança dos dados atualizados

- Dado que o QA ou dev verifica o banco de dados,
- Quando o usuário altera sua senha,
- Então a nova senha deve estar armazenada de forma segura (criptografada).

7 USER STORIES DESCONTINUADAS OU INCORPORADAS

Essas User Stories foram descontinuadas para evitar duplicidade de escopo e melhorar a clareza funcional. Seus conteúdos foram considerados e, quando aplicável, absorvidos em outras histórias mais abrangentes.

7.1 US #00: OBTENÇÃO DE RESPOSTAS DAS LLMS

User Story:

"Como usuário, eu quero ter as respostas das duas LLMS, para poder avaliá-las em critérios."

STATUS: *DESCONTINUADA – Absorvida pela US#02*

Justificativa: Em conversa com o Dev Team durante a Sprint 0, foi apontado que esta User Story já seria executada no andamento da US#02, por essa razão, seria redundante sua funcionalidade, por isso decidimos em conjunto por descontinua-lá.

7.2 US #23: PRESERVAÇÃO DO PROGRESSO AO RETORNAR PARA TELAS ANTERIORES

User Story:

"Como usuário, eu não quero perder meu progresso de avaliação ao voltar para telas anteriores, para que eu possa visualizar o que escolhi e verificar se errei em alguma avaliação."

Descrição:

Comentário PO:

– “Com a evolução da aplicação, percebi que esta User Story está intrinsecamente ligada à US:

A6S-28: Navegação para Telas Anteriores Durante a Avaliação) , sendo naturalmente atendida pela forma como a navegação e persistência de dados foram implementadas. Ainda assim, esta US foi documentada para preservar o histórico de decisões do backlog.”

STATUS: DESCONTINUADA – Absorvida pela US#23

Justificativa: Eliminado na Sprint 2 durante Sprint Planning pela duplicidade de funcionalidade com a US#21, por essa razão foi absorvida pela mesma.

7.3 US #09: GERAÇÃO DE RESPOSTAS BASEADAS NO CONTEXTO RECUPERADO

User Story:

"Como usuário, eu quero que a LLM utilize apenas o contexto recuperado para responder ao meu prompt, para que a resposta seja mais precisa e baseada em informações verificadas."

Descrição:

Comentário PO:

– “Essa US complementa a US08 (A6S-15: Enriquecimento das Respostas com Base Vetorizada - RAG Parte 1), e deve ser implementada após o enriquecimento via RAG estar funcional. A proposta aqui é criar um “modo controlado” de geração, onde a LLM não pode utilizar conhecimento genérico ou criativo fora do contexto retornado pela base vetorizada.

Pelo que pesquisei, essa restrição é especialmente útil para domínios sensíveis como saúde, e aproxima o sistema de um comportamento factual que reduz alucinações (IA opinando). Parâmetros como temperature e top_p também devem ser ajustados para garantir consistência.

Eu @Ivan Germano Rosendo de Oliveira Duarte, irei abrir uma conversa mais profunda sobre esse tema com o Cliente André Almeida, no dia 07/04/2025 (segunda-feira), para esclarecer se o nosso sistema RAG deve realmente ser mais “Restritivo” ao LLM, seguindo assim, a minha visão inicial.”

STATUS: DESCONTINUADA – Absorvida pela US#08

Justificativa: Esta User Story foi descontinuada durante o Sprint Planning da Sprint 2, após revisão conjunta do Dev Team e feedback do professor orientador. Constatou-se que a funcionalidade

descrita nesta US apresentava alta sobreposição com a US#08, diferenciando-se apenas em aspectos específicos que podem ser melhor tratados como subtasks ou critérios adicionais dentro da própria US#08. Além disso, a abordagem proposta nesta US#09 foi considerada muito restritiva e distante da proposta inicial do projeto, o que motivou ainda mais sua incorporação. Por esses motivos, a equipe optou por fundir a US#09 à US#08, visando manter o backlog mais coeso, evitar redundância e preservar o alinhamento funcional do sistema.

7.4 US #20: ALERTA DE ERRO NA OBTENÇÃO DAS RESPOSTAS

User Story:

"Como usuário, eu quero ser informado com mensagens de erro caso haja falha na obtenção das respostas, para que eu possa entender o problema e tentar novamente."

Descrição:

Comentário PO:

– “Esta User Story trata do tratamento de erros na resposta do servidor das LLMs. Caso ocorra uma falha, como uma LLM estar offline ou uma interrupção na rede, o sistema deve exibir uma mensagem clara ao usuário, explicando se a falha foi na comunicação com a LLM ou devido a problemas locais de conexão do próprio usuário.”

STATUS: DESCONTINUADA – Absorvida pela US#19

Justificativa: Esta User Story foi descontinuada durante o Sprint Planning da Sprint 2, após análise do Dev Team, por apresentar forte sobreposição com os critérios de aceitação e cenários de teste da US#19. Sua manutenção como história independente quebraria o princípio de independência do framework INVEST, além de introduzir redundância funcional no backlog. Dado que os comportamentos esperados da US#20 poderiam ser naturalmente tratados dentro do escopo da US#19, especialmente no contexto de erros de comunicação e timeout com as LLMs, a equipe optou por fundir ambas as histórias. Dessa forma, mantém-se a clareza do backlog e o foco em entregas mais completas e coesas.

8 CONSIDERAÇÕES FINAIS

O início da Sprint 3 marca o ciclo final do desenvolvimento da solução de avaliação de LLMs com feedback humano - AUXIA, e este documento representa um passo importante na organização do trabalho. Com as User Stories detalhadas, critérios de aceite e cenários definidos, a equipe possui uma base sólida para iniciar o desenvolvimento com segurança e alinhamento. Este backlog será revisado e evoluído continuamente, acompanhando as descobertas e aprendizados que surgirem ao longo do projeto.

Elaborado por:

Duarte I. G. R. O. – Product Owner – API 6º ADS

Análise e Desenvolvimento de Sistema – FATEC Profº Jesen Vidal

Data: 12/05/2025

Assinatura: _____

São José dos Campos, 12 de maio de 2025