ANEXO 5 - INTEGRAÇÃO COM A PLATAFORMA DE CHAT INSTITUCIONAL COM IA GENERATIVA DA UFF

A CONTRATADA fica obrigada a montar um planejamento arquitetural, implementar e integrar na sua solução de ESM (Enterprise Service Manager) de forma embarcada e ou integrada de forma nativa e ou por meio de API RESTful & Webhooks ou outro meio de integração de formulários por meio de um centro de comando e controle da IA generativa da UFF com a solução de IA generativa de mercado e ou da própria solução, de forma que todas as informações sejam roteadas e integradas por um único barramento de forma que as informações sejam conectadas, integradas e conciliadas dentro de um único framework.

Contextualização da Solução de Chat Institucional e do Projeto da Central de Atendimento Única (CAU). Este documento estabelece os requisitos técnicos para a integração entre a futura solução de Gerenciamento de Serviços de TI (ITSM) a ser contratada e a solução de Chat Institucional apoiada por Inteligência Artificial Generativa, desenvolvida internamente pela Superintendência de Tecnologia da Informação (STI) da Universidade Federal Fluminense (UFF).

1.1. Contexto Institucional, problematização e iniciativas em desenvolvimento

A Universidade Federal Fluminense (UFF), presente em 9 municípios do Estado do Rio de Janeiro e com estrutura que abrange mais de 150 mil membros da comunidade acadêmica, enfrenta desafios significativos quanto à dispersão e inconsistência de seus canais de atendimento. A universidade possui uma estrutura organizacional de grande porte e complexidade, com 42 Unidades Acadêmicas, mais de 120 cursos de graduação e 81 programas de pós-graduação, distribuídos por 9 municípios. Atende a uma comunidade de aproximadamente 7,5 mil servidores e mais de 105 mil alunos, além do público externo.

Atualmente, exceto pelos serviços de TI providos pela Superintendência de Tecnologia da Informação (STI), os demais serviços são prestados por meio de e-mails e sites isolados, através de ligação telefônica ou de forma presencial , sem centralização de informações, o que gera dificuldade de acesso, ineficiência operacional e fragmentação institucional.

Nesse cenário, a Superintendência de Tecnologia da Informação da UFF está liderando o desenvolvimento de uma Central de Atendimento Única (CAU), com o objetivo de estabelecer um ponto central de contato entre a comunidade e as áreas provedoras de serviços. Essa iniciativa visa à padronização, agilidade, medição de desempenho e conveniência no atendimento, integrando áreas acadêmicas, administrativas e o público externo.

A CAU será apoiada por uma solução de chat institucional com IA generativa, baseada na arquitetura RAG (Retrieval-Augmented Generation). Essa arquitetura combina mecanismos de busca vetorial e léxica com modelos de linguagem natural para responder perguntas dos usuários. A solução utiliza fontes como Sites da UFF, Bases de Conhecimento (SEI, Citsmart), Boletins de Serviço e o RAD (Relatório Anual de Docentes), que são periodicamente processados, segmentados e vetorizados semanticamente. A IA generativa, operando com modelos como OpenAI, LLaMA e Deepseek, utiliza esse banco vetorial para construir respostas contextualizadas, seguras (com uso de guardrails), e com reclassificação, para garantir maior precisão.

1.2. Arquitetura da Solução de Chat Institucional com IA Generativa

Como pilar tecnológico de suporte à CAU, a STI está desenvolvendo uma solução de Chat Institucional baseada em Inteligência Artificial Generativa, utilizando a arquitetura Retrieval-Augmented Generation (RAG). De forma não técnica, a solução funciona da seguinte maneira:

1. Ingestão de Dados: Um processo automatizado e agendado coleta, trata e segmenta informações de fontes de dados oficiais e confiáveis da UFF, como:

○ Sites do domínio uff.br;

○ Base de Conhecimento do sistema CitSmart (legado);

○ Base de Conhecimento do sistema SEI;

○ Boletins de Serviço e Relatórios Anuais Docentes (RAD).

2. Busca e Recuperação: Quando um usuário envia uma pergunta, o sistema realiza uma busca híbrida (léxica e semântica) em um banco de dados vetorial para encontrar os trechos de documentos mais relevantes para responder àquela questão.

3. Geração da Resposta: A IA Generativa (utilizando modelos como Llama, Deepseek, OpenAI entre outros) recebe a pergunta original do usuário e o contexto recuperado (os trechos dos documentos) para, então, sintetizar uma resposta coesa, precisa e em linguagem natural.

4. Entrega e Segurança: A resposta é validada por mecanismos de segurança ("Guardrails") para evitar conteúdo inadequado e, em seguida, entregue ao usuário através de uma interface de chat.

Esta arquitetura garante que as respostas não sejam "inventadas", mas sim fundamentadas em documentos e informações oficiais da instituição, aumentando a confiabilidade e a precisão do atendimento.

1.3. Status Atual do Projeto (Base: Fevereiro de 2025)

O projeto da CAU encontra-se em fase avançada de desenvolvimento e implantação, atuando em duas frentes principais:

● Operação da Central Única: Já foram mapeados e implantados os primeiros serviços no portal de atendimento, começando pela Superintendência de Documentação (SDC), com foco no Repositório Institucional (RiUFF). Os fluxos, SLAs e treinamentos para esta área estão em andamento.

● Autoatendimento com IA Generativa: A frente de IA está igualmente madura, com as fontes de dados definidas, os mecanismos de extração e vetorização implementados, e um protótipo funcional de chat (web e integrado a plataformas como Telegram) em operação para testes. Estudos avançados, como a consulta a bancos de dados estruturados (Text-to-SQL), também estão em andamento, demonstrando a capacidade técnica da solução.

Dada a maturidade de ambas as frentes, é imperativo que a nova ferramenta de ITSM a ser contratada possua capacidade nativa de se integrar à solução de IA da UFF.

2. Requisitos de Integração para a Ferramenta de ITSM

Para garantir a sinergia entre a ferramenta de ITSM contratada e o Chat Institucional da UFF, a proponente deverá fornecer uma solução que atenda aos seguintes requisitos de integração.

2.1. Premissas Gerais de Integração

A integração deverá ser realizada por meio de uma interface de programação de aplicações (API) baseada no padrão REST (Representational State Transfer). A API da solução contratada deve ser pública, bem documentada, estável e abranger as funcionalidades listadas abaixo.

A autenticação e autorização entre os sistemas deverão ser garantidas pelo uso de API Keys ou método de segurança superior (como OAuth 2.0), assegurando o reconhecimento mútuo e a comunicação segura para

todas as requisições.

Funcionalidades da API (Endpoints)

A API da ferramenta de ITSM deverá prover, no mínimo, as seguintes funcionalidades acessíveis pela solução de Chat da UFF:

● *Endpoint 1: Consulta ao Chat Institucional (Chamada da ITSM para o Chat UFF)*

○ *Descrição: A ferramenta ITSM deve ser capaz de realizar uma chamada via API para a solução de Chat da UFF, enviando uma pergunta de um usuário. Isso permitirá, por exemplo, que um "widget" de ajuda dentro do portal de serviços da ITSM seja alimentado diretamente pela IA da UFF.*

○ *Método (Exemplo): POST /ask*

○ *Input: Pergunta do usuário em formato texto (e.g., JSON com um campo {"question": "Como solicito meu diploma?"}).*

○ *Output: Resposta gerada pela IA em formato texto (e.g., JSON com um campo {"answer": "..."}).*

● *Endpoint 2: Consulta de Chamados por Usuário (Chamada do Chat UFF para a ITSM)*

○ *Descrição: A solução de Chat da UFF deve poder consultar a lista de chamados (abertos, em andamento, etc.) de um determinado usuário, para informá-lo sobre o status de suas solicitações.*

○ *Método (Exemplo): GET /tickets?user_id={identificador_do_usuario}&status={em_andamento}*

○ *Input: Um identificador único do usuário (CPF, idUFF, etc.) e status do chamado (aberto, em andamento etc).*

○ *Output: Uma lista de chamados associados ao usuário, contendo no mínimo:*

ID do chamado, título, status e data de abertura.

● *Endpoint 3: Criação de Chamado (Chamada do Chat UFF para a ITSM)*

○ *Descrição: O Chat UFF deve ser capaz de realizar uma chamada via API para a solução ITSM, enviando o identificador do usuário e a descrição do chamado. Isso permitirá que o usuário abra um chamado diretamente pelo Chat UFF. A solução ITSM deve obrigatoriamente retornar o número do chamado.*

○ *Método (Exemplo): POST /create*

○ *Input: descrição do problema do usuário em formato texto, identificador do usuário (e.g., JSON com um campo {"identificador_do_usuario": "123456", "description": "Estou com problema para solicitar meu diploma."}).*

○ *Output: Número do chamado gerado (e.g., JSON com um campo {"identificador_do_chamado": "98765"}).*

● *Endpoint 4: Consulta de Chamado Específico (Chamada do Chat UFF para a ITSM)*

○ *Descrição: A solução de Chat da UFF deve poder obter os detalhes completos de um chamado específico, mediante o seu identificador.*

○ *Método (Exemplo): GET /tickets/{identificador_do_chamado}*

○ *Input: O ID do chamado.*

○ *Output: Um objeto JSON com todas as informações relevantes do chamado, como título, descrição completa, status, histórico de interações, solicitante, responsável, etc.*

● *Endpoint 5: Consulta à Base de Conhecimento do ITSM (Chamada do Chat UFF para a ITSM)*

○ *Descrição: Funcionalidade crítica para a arquitetura RAG. A solução de Chat da UFF deve poder consultar e extrair o conteúdo de artigos da base de conhecimento da ferramenta ITSM. Isso permitirá que o conhecimento registrado na nova plataforma alimente dinamicamente a IA.*

○ *Método (Exemplo): GET /knowledge_base/articles/{identificador_do_artigo}*

○ *Input: O ID do artigo na base de conhecimento.*

○ *Output: O conteúdo completo do artigo, preferencialmente em formato estruturado (JSON ou HTML limpo), contendo título, corpo do texto, anexos, etc. Deverá haver também um endpoint para listar todos os artigos da base para permitir a indexação periódica.*

A empresa deverá entregar um projeto de implementação e integração em até 12 (doze) dias úteis após a UFF entregar para a empresa CONTRATADA a IA Generativa da UFF, sua arquitetura, detalhamento da infraestrutura, do código e da modelagem do RAG de forma com que seja possível a CONTRATADA entregar um PROJETO ARQUITETURAL e um planejamento de implementação, conforme a especificação a seguir:

1. Objetivo

Garantir a plena integração da solução de Inteligência Artificial Generativa baseada em RAG ao sistema de gestão de serviços corporativos (ESM) a ser contratado, de modo que essa IA atue como um componente nativo da solução, acessando e inferindo conhecimento de forma transparente ao usuário final e aos agentes de atendimento, com total aderência às interfaces técnicas modernas (API, webhook, webhook inverso, etc).

---

2. Requisitos Arquiteturais

2.1 Arquitetura Modular e Acoplável

A solução ESM deverá:

- *Ser agnóstica quanto ao motor de IA a ser integrado, permitindo plugar e gerenciar instâncias externas de IA generativa via configurações, APIs ou conectores padrão.*
- *Suportar componentização e orquestração de serviços, permitindo embarcar a IA generativa como um componente embutido, sem que o usuário final perceba disjunção entre módulos.*
- *Permitir customização do front-end (portal de serviços, chatbots, etc) com componentes UI embutidos e interativos que façam chamadas contextuais à IA.*

2.2 Suporte a Integração RAG

O ESM deverá:

- *Permitir que a IA generativa consulte bases de conhecimento internas e externas via APIs.*
- *Suportar integração com vetores semânticos oriundos de bases como documentos institucionais, FAQs, repositórios técnicos, sistemas de ticketing históricos, etc.*
- *Expor um pipeline de dados estruturado (via GraphQL, RESTful APIs, Webhook, Websocket ou outra tecnologia compatível com eventos) para que a solução RAG possa indexar o conteúdo necessário à geração aumentada por recuperação.*

---

3. Requisitos de Integração (nível técnico)

3.1 API RESTful & Webhooks

- *A solução ESM deverá expor APIs RESTful documentadas com suporte a:*
  - *Autenticação OAuth2.0 ou JWT.*
  - *Operações CRUD completas em tickets, usuários, KB, SLAs, workflows.*
  - *Disparo de eventos via webhooks configuráveis (onTicketCreate, onSearchFail, onPortalSearch, etc).*
  - *Envio de contexto conversacional para o endpoint da IA generativa (em formato JSON).*

3.2 Webroots e Endpoint Abstrato

- *Suporte à criação de webroots customizados, permitindo encaminhar queries do portal diretamente para endpoints da IA generativa (por exemplo: https://servicos.uff.br/ia/consultar).*
- *Configuração de reverse proxy (caso necessário) para preservar identidade de domínio e cookies de sessão UFF.*

3.3 Suporte a Plugins e Middlewares

- *A solução deverá permitir inserção de middlewares próprios, permitindo:*
  - *Pré-processamento de chamadas do usuário.*
  - *Injeção de resultados da IA antes da renderização final.*
  - *Encaminhamento de intents para execução automatizada de tickets ou ações (RPA-like).*

---

4. Funcionalidades Esperadas da IA Integrada

4.1 Autoatendimento e Geração de Respostas

- *A IA deverá:*
  - *Ser acionada em consultas no portal de serviços ou via chatbot.*
  - *Utilizar a arquitetura RAG para consultar fontes internas atualizadas (documentação, legislações, base de tickets, manuais).*
  - *Retornar respostas completas, auditáveis, com links para fontes e possibilidade de feedback do usuário.*

4.2 Apoio a Analistas de Atendimento

- *A IA deverá ser capaz de:*
  - *Responder internamente ao analista (modo copilot).*
  - *Sugerir preenchimento de chamados, categorias, soluções e scripts com base no conteúdo.*
  - *Executar ações automatizadas via orquestração ESM + API IA (ex: abrir tíquete, consultar SLA, sugerir workflow).*

4.3 Auditoria e Governança

- *Toda interação com a IA deverá:*
  - *Ser registrada em log técnico.*
  - *Suportar análise de segurança, anonimização e LGPD.*
  - *Ser configurável via parâmetros de governança técnica (ex: thresholds de confiança, fallback para humano).*

---

5. Compatibilidade e Escalabilidade

- *A solução deverá funcionar com infraestrutura híbrida ou cloud-native, permitindo:*
  - *Execução da IA generativa em cloud (Azure OpenAI, AWS Bedrock, HuggingFace, etc).*
  - *Indexação e armazenamento vetorial interno, com base em ferramentas como:*
    - *FAISS, Weaviate, Pinecone, Qdrant, Milvus.*
  - *Escalabilidade horizontal com balanceamento de carga.*

---

6. Padrões de Mercado e Protocolos Suportados

- *OpenAPI (Swagger) 3.x*
- *OAuth2.0 / OIDC*
- *Webhooks bidirecionais*
- *JSON-LD / JSON Schema para ontologias*
- *GraphQL (desejável)*
- *MQTT / Kafka para eventos em tempo real (opcional)*

---

7. Casos de Uso Obrigatórios

1. *Busca no portal: Usuário final pesquisa por "segunda via do diploma"; IA responde com conteúdo baseado na base vetorial e links oficiais.*
2. *Sugestão automática: Agente abre tíquete de "revalidação de diploma estrangeiro" e IA sugere texto padrão + links + workflow correspondente.*
3. *Criação de tíquete por IA: Usuário descreve sua necessidade ("não consigo acessar o SIGA") e a IA cria automaticamente o tíquete completo com categoria e encaminhamento.*

4. *Chatbot contextual: Interação com IA pelo chatbot, com memória de conversação e referência aos documentos da UFF.*
5. *Geração de relatórios com base em requisições: Analista solicita "quais os serviços mais solicitados em 2024 com tempo médio acima do SLA?" e IA gera um relatório pré-analisado.*