Exploração de Informação de Ativos Industriais no Padrão Asset Administration Shell via Modelos de Linguagem de Grande Escala

Felipe Lima de Medeiros
Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte
Natal, Brasil
felipe.lima.092@ufrn.edu.br

Gustavo Bezerra Paz Leitão
Instituto Metrópole Digital (IMD)
Universidade Federal do Rio Grande do Norte
Natal, Brasil
gustavo.leitao@imd.ufrn.br

Luiz Affonso Guedes
Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte
Natal, Brasil
affonso@dca.ufrn.br

***Abstract*—The Asset Administration Shell (AAS) is a standard for the digital representation of industrial assets, consolidating information and functionalities in a unified format to facilitate system integration. However, its technical complexity hinders access and exploration of this data by users without specialized knowledge.**

This work proposes a conversational interface developed with the LangChain4j library, which connects the system to large language models (LLMs) and can interpret natural language queries to retrieve information from AAS files. The solution aims to democratize information access and lower the entry barrier for new users. Experiments were conducted using three types of questions: direct, relational, and interpretative, applied to two groups of files with varying complexity levels. The models GPT-4o mini and GPT-4.1 mini were used to evaluate both response accuracy and execution time. Results demonstrate the effectiveness of the approach in information extraction, particularly in scenarios with fewer assets. Additionally, the proposed architecture is scalable and flexible, adapting to advances in language models and expanding application possibilities.

***Keywords*—Asset Administration Shell, AAS, Industry 4.0, large language models**

***Resumo*—O Asset Administration Shell (AAS) é um padrão que representa digitalmente ativos industriais, reunindo informações e funcionalidades em um formato unificado para facilitar a integração entre sistemas. Contudo, sua complexidade técnica dificulta o acesso e a exploração dessas informações por usuários sem conhecimento especializado.**

Este trabalho propõe uma interface conversacional, desenvolvida com a biblioteca LangChain4j, que conecta o sistema aos grandes modelos de linguagem (LLM) e interpreta perguntas em linguagem natural para consultar arquivos no padrão AAS. A solução visa democratizar o acesso às informações e reduzir a barreira de entrada para novos usuários. Foram realizados experimentos com três tipos de perguntas — diretas, relacionais e interpretativas — aplicados a dois grupos de arquivos com diferentes níveis de complexidade. Utilizaram-se os modelos `GPT-4o mini` e `GPT-4.1 mini`, avaliando tanto a precisão das respostas quanto o tempo de execução. Os resultados mostram a eficácia da abordagem na extração de informações, especialmente em cenários com menor volume de ativos. Além disso, a arquitetura proposta é escalável e flexível, acompanhando a evolução dos modelos de linguagem e ampliando as possibilidades de aplicação da solução.

***Palavras-Chave*—Asset Administration Shell, AAS, Indústria 4.0, large language models**

I. Introdução

A Indústria 4.0 demanda a digitalização dos ativos industriais para promover a modernização e a inovação, com a utilização de modelos digitais precisos e confiáveis. Os dados gerados por esses ativos tornam-se recursos estratégicos essenciais para a tomada de decisões, sendo a padronização um requisito fundamental para seu uso eficaz [1]. No entanto, a heterogeneidade dos ativos representa um desafio significativo para a integração eficiente dos sistemas industriais.

Para enfrentar essas dificuldades, foi desenvolvido o *Asset Administration Shell* (AAS) [2], uma especificação para a representação digital de ativos físicos ou digitais [3]. Inserido na arquitetura da Indústria 4.0, o AAS organiza informações de forma padronizada e que representa precisamente as características e propriedades dos ativos reais, promovendo interoperabilidade, modularidade e reutilização.

Embora o AAS ofereça grande flexibilidade ao representar uma ampla variedade de ativos, essa abordagem genérica também contribui para sua complexidade. Isso dificulta a compreensão do padrão, especialmente para profissionais que não estão familiarizados com seus detalhes técnicos. Assim, sua complexidade técnica pode representar uma barreira significativa para sua adoção em larga escala, exigindo equipes especializadas para sua implementação e manutenção.

Alguns estudos têm abordado esse desafio, propondo soluções para simplificar a complexidade estrutural do AAS e viabilizar sua adoção no contexto industrial, por meio de abordagens inovadoras que facilitam a geração dos arquivos

AAS, como demonstram trabalhos recentes [4], [5]. Esses métodos desempenham um papel crucial na promoção do AAS e na facilitação de sua adoção em larga escala.

Apesar das iniciativas existentes, ainda há lacunas significativas na exploração, compreensão e interpretação dos ativos representados pelo AAS. Ferramentas como o *AASX Explorer* [6], que oferecem uma visualização hierárquica do conteúdo, têm limitações na interação com os dados. A abordagem hierárquica, embora funcional, exige uma navegação manual extensa, o que dificulta a usabilidade.

Para enfrentar essas limitações de usabilidade, este artigo propõe uma solução baseada em modelos de linguagem de grande escala (LLMs). Esses modelos têm se mostrado eficazes em uma vasta gama de contextos, como na interação com bases de conhecimento complexas [7] e na automação de tarefas técnicas específicas, como a geração de código a partir da documentação [8]. A versatilidade dos LLMs também inclui sua aplicação em setores como saúde, educação, direito e finanças [9], além de possibilitar a criação de interfaces conversacionais que permitem interações naturais com os usuários [10], tornando-os uma tecnologia estratégica para explorar estruturas digitais complexas como o AAS.

Ao disponibilizar uma interface conversacional baseada em LLM, é possível simplificar substancialmente a consulta e a interpretação sobre os ativos, permitindo que operadores e engenheiros sem conhecimento técnico especializado extraiam informações detalhadas sobre ativos industriais de forma intuitiva. A habilidade dos LLMs de compreender estruturas complexas e fornecer respostas precisas de forma acessível oferece uma camada de abstração que elimina a necessidade de conhecimento especializado sobre formatos de dados ou linguagens de consulta. Isso torna a navegação e a extração de informações mais simples e eficientes, reduzindo a curva de aprendizado necessária para utilizar plenamente o padrão AAS.

A proposta visa, portanto, aprimorar a usabilidade do AAS, facilitando o acesso aos dados armazenados em suas estruturas complexas e incentivando a adoção do padrão em ambientes industriais. Ao tornar a tecnologia mais acessível e intuitiva, busca-se expandir seu uso no setor industrial, contribuindo para a disseminação de uma infraestrutura digital padronizada, interoperável e de fácil acesso, alinhada aos princípios fundamentais da Indústria 4.0.

O restante do artigo está estruturado da seguinte forma: a Seção II apresenta o conceito de ativo industrial e resume o padrão AAS; a Seção III discute os principais desafios na adoção e compreensão da especificação AAS; a Seção IV descreve a solução proposta, que oferece um modo de interação em linguagem natural para explorar estruturas AAS; a Seção V descreve os experimentos realizados e analisa os resultados obtidos; finalmente, a Seção VI apresenta a conclusão e sugestões para trabalhos futuros.

II. Ativos industriais e o Padrão Asset Administration Shell

Esta seção apresenta os conceitos fundamentais relacionados aos ativos industriais e ao padrão Asset Administration Shell (AAS). Primeiramente, define-se o conceito de ativo industrial, destacando sua relevância e a importância da digitalização para a otimização de processos. Em seguida, discute-se o padrão AAS, que oferece uma estrutura padronizada para representar os ativos de maneira integrada e acessível, promovendo a interoperabilidade entre sistemas no ambiente industrial.

A. Ativo industrial

Um ativo industrial é um recurso de propriedade de uma empresa utilizado na produção ou entrega de bens e serviços. Esses ativos incluem máquinas, edifícios, equipamentos, veículos e sistemas tecnológicos essenciais para a fabricação, processamento e outras atividades industriais.

A gestão de ativos industriais é um tópico de grande interesse no setor industrial. Este foco não se limita apenas aos ativos financeiros, mas também compreende ativos físicos, humanos, informacionais e intangíveis [11].

Uma boa gestão de ativos permite que uma organização crie, aprimore e sustente seus negócios, pois pode aumentar a produtividade econômica [12] e permite que a organização economize dinheiro a longo prazo [13].

A digitalização dos ativos industriais através de tecnologias como a Internet das Coisas (IoT) aprimora a capacidade de monitoramento, gestão e otimização de processos. A diversidade de termos empregados, como IoT, Internet Industrial, Sistemas Ciberfísicos e Gêmeos Digitais, reflete o interesse e a complexidade do tema. Embora esses conceitos tenham escopos ligeiramente sobrepostos, cada um descreve aplicações e funcionalidades distintas. Contudo, o objetivo central é a integração e a interoperabilidade eficazes de dispositivos, serviços e fontes de dados industriais. Para que as implementações práticas sejam bem-sucedidas, é essencial definir claramente os formatos de dados, as interfaces e o significado semântico dos objetos e atributos referenciados [14].

Para isso, foi desenvolvida a especificação Asset Administration Shell (AAS), com o intuito de descrever um ativo industrial de forma flexível e genérica, seguindo um padrão bem definido.

B. Asset Administration Shell (AAS)

O Asset Administration Shell (AAS) é uma representação digital dos ativos reais [15]. Ele funciona como uma abstração digital que armazena todas as informações importantes sobre um ativo. Foi criado para representar qualquer ativo, incluindo materiais, produtos, dispositivos, máquinas e também software e serviços digitais [3]. O AAS tem como objetivo criar uma representação digital padronizada dos ativos, garantindo a interoperabilidade e facilitando a troca de informações entre diferentes sistemas e partes interessadas no contexto da Indústria 4.0.

Essa especificação é composta por dois componentes principais: o modelo de informação e o protocolo de comunicação. O modelo de informação descreve a estrutura e o conteúdo dos dados usados para representar um ativo, enquanto o protocolo de comunicação estabelece como esses dados são transmitidos.

Este trabalho concentra-se especificamente no modelo de informação do AAS, com o objetivo de propor uma abordagem que simplifique a exploração de sua estrutura, sem a necessidade de ter um profundo conhecimento dessa especificação.

C. Metamodelo AAS

A especificação do metamodelo do AAS [2] pode ser explicada separando em duas partes: header (cabeçalho) e body (corpo). O header representa informações gerais, enquanto o body detalha as propriedades do ativo. A figura 1 apresenta um resumo da organização do metamodelo AAS, e logo em seguida há uma explicação sobre sua estrutura.

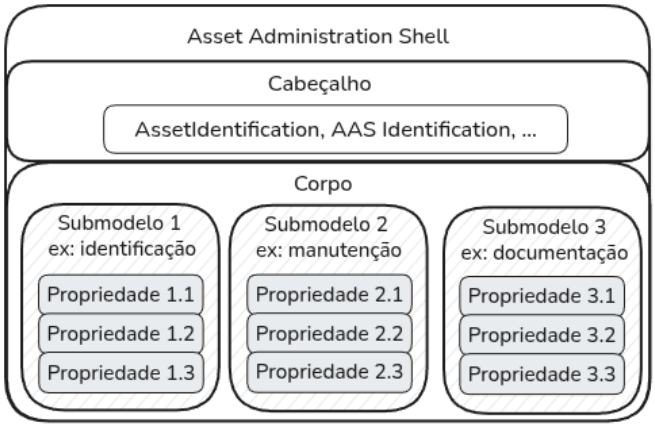

Fig. 1. Estrutura do AAS

1) Cabeçalho: O cabeçalho do modelo AAS inclui uma lista de propriedades, como a identificação do ativo e a identificação única do próprio AAS. Essas informações devem ser inseridas para designar claramente a identidade exclusiva do componente [15]. O cabeçalho pode ser visto como um conjunto definido de informações que fornecem detalhes sobre a identificação do ativo.

2) Corpo: O corpo contém um conjunto de submodelos que consiste em propriedades hierarquicamente bem organizadas, vinculadas a propriedades ou funções.

Um submodelo ajuda a estruturar o AAS em componentes distintos, cada um abordando um aspecto específico da representação ou da funcionalidade do ativo. Isso facilita a gestão e a reutilização do AAS, tornando-o mais modular e flexível.

Cada submodelo está associado a um domínio ou tópico específico e é composto por um conjunto de propriedades. Como ilustrado na figura 1, o ativo neste exemplo é representado por três submodelos: identificação, manutenção e documentação. Cada submodelo possui suas próprias propriedades e opera de forma independente. A combinação desses submodelos e suas respectivas propriedades resulta na representação completa do ativo.

Essa abordagem destaca a flexibilidade na representação de um ativo, permitindo que o modelo de dados seja versátil e genérico. Assim, ele pode incluir e representar a diversidade de ativos existentes.

III. Caracterização do Problema

O *Asset Administration Shell* (AAS) surge como um padrão promissor para a representação digital de ativos na Indústria 4.0, mas é notável a presença de alguns entraves que podem comprometer sua adoção prática.

A. Heterogeneidade dos ativos

A grande variedade de ativos — desde sensores e atuadores até máquinas complexas e serviços digitais — resulta em modelos AAS com estruturas e propriedades muito diferentes entre si. Essa heterogeneidade dificulta a criação de processos de consulta e análise, pois cada instância do AAS pode apresentar formatos e vocabulários próprios.

B. Complexidade do modelo de informação

O modelo de informação do AAS foi concebido de forma genérica e extensível, o que exige conhecimento técnico aprofundado para compreender e navegar em suas camadas de abstração. Documentos de especificação extensos e terminologia especializada aumentam a curva de aprendizado, obrigando o usuário a investir tempo na assimilação de conceitos antes mesmo de extrair dados relevantes.

C. Limitações das ferramentas atuais

As ferramentas utilizadas atualmente para exploração de arquivos AAS, como o *AASX Explorer* [6], apresentam limitações significativas do ponto de vista de usabilidade. A interface, baseada em uma visualização hierárquica semelhante a uma estrutura de pastas, embora útil para inspeção técnica detalhada, torna a navegação lenta e pouco intuitiva, especialmente quando aplicada a ativos com estruturas complexas ou múltiplos submodelos.

Esse formato de exploração exige que o usuário percorra manualmente diversos níveis da hierarquia para localizar uma informação específica, o que é particularmente problemático em arquivos extensos, que podem conter milhares de linhas em JSON. O JSON, como formato leve de intercâmbio de dados [16], foi projetado para ser legível por humanos e fácil de ser analisado por máquinas [17]. No entanto, em arquivos grandes, tarefas simples, como obter um resumo geral do ativo ou acessar propriedades pontuais, tornam-se trabalhosas e exigem conhecimento prévio da estrutura do modelo.

Essas limitações evidenciam a necessidade de soluções mais acessíveis e centradas no usuário, que permitam uma interação mais direta e natural com os dados dos ativos digitais.

D. Barreiras à adoção e manutenção

A conjugação de heterogeneidade, complexidade e falta de usabilidade causa a necessidade de equipes especializadas para implementar, manter e consultar AASs. Essa dependência de expertise técnica pode limitar a adoção em larga escala do padrão, especialmente em organizações com recursos de TI restritos ou sem experiência prévia com soluções de automação avançada.

Em face desses desafios, identifica-se a demanda por uma abordagem que auxilie o usuário na exploração do AAS de forma mais acessível e eficiente.

IV. Solução Proposta

A solução proposta neste trabalho visa facilitar a usabilidade entre os usuários e os ativos no padrão AAS, superando as barreiras impostas pela complexidade de sua estrutura e pela necessidade de conhecimento técnico especializado. Para isso, foi desenvolvido um sistema web que permite a exploração e consulta de dados AAS utilizando linguagem natural, mediada por modelos de linguagem de grande escala (LLMs).

A. Arquitetura do sistema

O backend foi implementado em *Java*, utilizando o framework *Spring*, que oferece uma estrutura robusta para construção de aplicações web modernas, com suporte a injeção de dependência, controle de rotas e integração com serviços externos. O armazenamento dos dados estruturados provenientes dos arquivos AAS é realizado em um banco de dados relacional *PostgreSQL*, escolhido por sua confiabilidade e compatibilidade com o ecossistema Java.

Para o desenvolvimento do frontend, foi utilizado o framework *Thymeleaf*, que permite a renderização dinâmica de templates HTML no lado do servidor, integrado de forma fluida com o backend em Spring. O Thymeleaf oferece uma abordagem simples para gerar interfaces ricas e interativas, facilitando a visualização dos dados AAS de forma acessível e intuitiva para o usuário.

Para permitir a comunicação entre a aplicação e os modelos de linguagem, foi utilizada a biblioteca *LangChain4j* [18], uma adaptação da popular LangChain para o ambiente Java, que simplifica a construção de soluções de consulta em linguagem natural ao possibilitar a orquestração de prompts, o gerenciamento de contexto e a integração com diferentes provedores de LLMs. Um dos benefícios dessa arquitetura é a facilidade na substituição ou atualização do modelo de linguagem utilizado, o que garante flexibilidade à solução. Foram utilizados dois modelos da **OpenAI**: o **GPT-4o mini**, escolhido pelo seu bom equilíbrio entre desempenho e custo computacional, e o **GPT-4.1 mini**, empregado para avaliar o impacto de modelos mais robustos na precisão e profundidade das respostas.

Embora nesta implementação tenham sido utilizadas LLMs externas, acessadas via API da OpenAI, com fins experimentais e em ambiente controlado, vale destacar que, em contextos industriais, o uso de modelos de linguagem pode demandar requisitos mais rigorosos de segurança e privacidade. Nesses cenários, especialmente quando há acesso a dados sensíveis, pode-se considerar alternativas mais controladas, como a execução de LLMs em ambientes privados, seja em infraestrutura on-premise ou em nuvens com controle dedicado e ambiente isolado. A arquitetura proposta é compatível com essas abordagens, permitindo sua adaptação a diferentes níveis de exigência.

A figura 2 resume a arquitetura da solução proposta. Essa combinação de tecnologias permite que usuários possam interagir com os dados dos ativos digitais sem necessidade de conhecimento prévio sobre o padrão AAS, apenas por meio de perguntas e comandos em linguagem natural. O sistema interpreta as intenções do usuário, extrai as informações relevantes da base de dados e retorna respostas claras, contextualizadas e úteis, promovendo acessibilidade e eficiência na adoção de soluções baseadas em AAS.

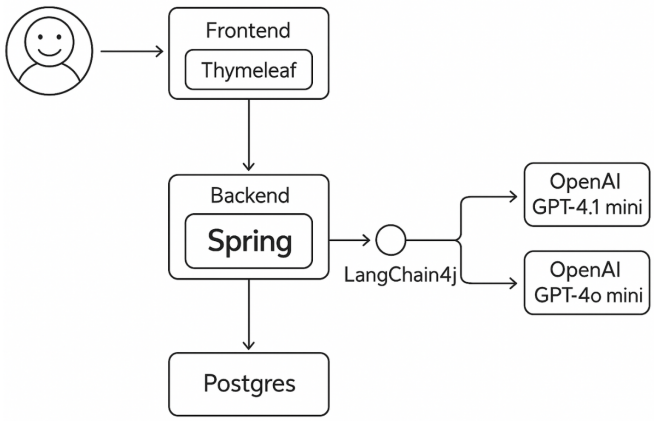

Fig. 2. Arquitetura da solução proposta

B. Funcionalidades

A seguir, são apresentadas as principais funcionalidades da solução proposta:

1) Cadastro de Grupo: No sistema desenvolvido, o conceito de *grupo* foi adotado como uma estratégia de organização lógica dos arquivos AAS, permitindo ao usuário estruturar os ativos de acordo com suas próprias necessidades e critérios. Cada grupo pode conter um ou mais arquivos nos formatos `.aasx` ou `.json`. Essa flexibilidade visa facilitar o uso da ferramenta em diferentes contextos industriais.

Por exemplo, um usuário pode criar um grupo denominado *Sensores de Temperatura* e incluir nele todos os arquivos AAS correspondentes a sensores desse tipo, enquanto outro grupo pode reunir ativos pertencentes a uma linha de produção específica. Essa abordagem permite maior liberdade para a segmentação e o gerenciamento dos ativos, refletindo a diversidade de aplicações práticas no ambiente industrial. Na figura 3 é exibida a interface gráfica do sistema para o cadastro do grupo.

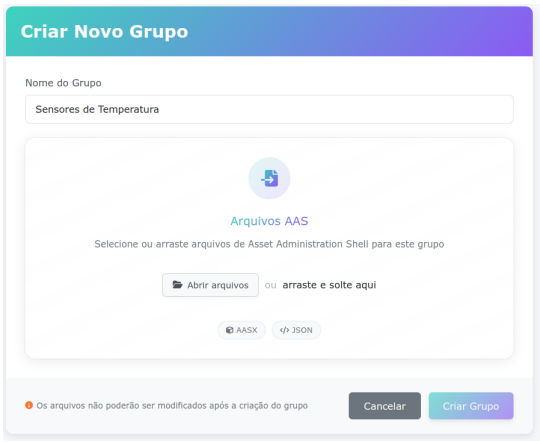

Fig. 3. Tela de cadastro de grupo

2) Visualização de Grupos: Após o cadastro, os grupos ficam disponíveis em uma listagem, conforme ilustrado na figura 4, organizados em formato de *cards*, permitindo o acesso individual a cada grupo.

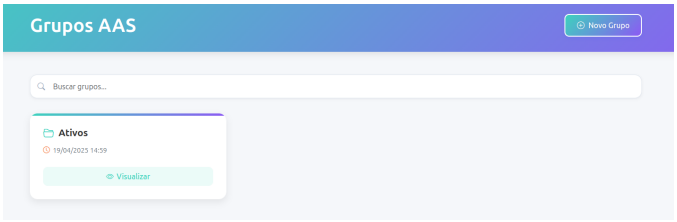

Fig. 4. Listagem dos grupos cadastrados

3) Exploração Estrutural do Grupo: Ao selecionar um grupo, o usuário é direcionado para uma página de *detalhamento*. Nessa tela, há duas formas de visualizar a estrutura do grupo:

- Visualização do JSON bruto, como na figura 5;
- Visualização hierárquica em formato de árvore (*tree view*), como mostrado na figura 6.

Fig. 5. Visualização no formato JSON do grupo AAS

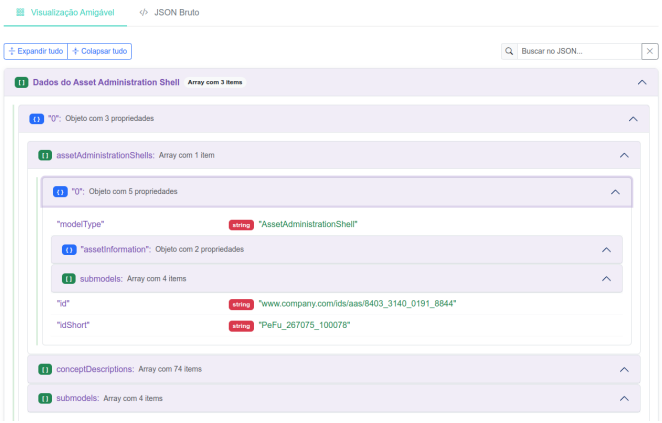

Fig. 6. Visualização hierárquica do grupo AAS

4) Chat com Modelo de Linguagem de Grande Escala: O diferencial da proposta está na possibilidade de interação por meio de um *chat especializado*, suportado por um modelo de linguagem de grande escala, como pode ser visto na figura 7. O usuário pode fazer perguntas em linguagem natural sobre o conteúdo do grupo selecionado. O sistema gera automaticamente o contexto necessário a partir da estrutura do grupo AAS armazenada e envia a requisição para o modelo via LangChain4j, retornando uma resposta textual clara e objetiva.

Internamente, os prompts foram projetados para restringir as respostas ao escopo do AAS do grupo em questão, aumentando a precisão e evitando interpretações indevidas. Essa estratégia reforça a segurança e a confiabilidade das respostas.

Essa solução torna a interação com o AAS significativamente mais acessível, permitindo que usuários com pouca familiaridade técnica obtenham informações de forma simples e direta. Ao eliminar a necessidade de navegação manual por estruturas complexas, o sistema contribui para uma experiência mais intuitiva e eficiente, promovendo a usabilidade e ampliando o acesso às tecnologias digitais no contexto da Indústria 4.0.

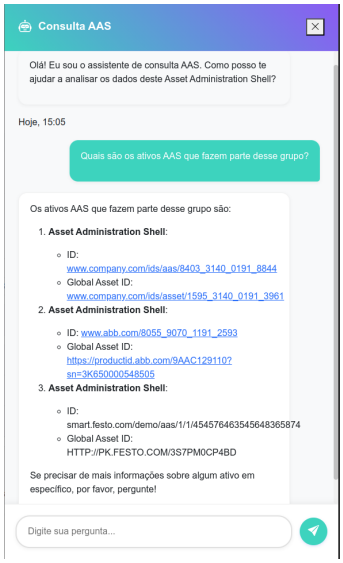

Fig. 7. Chat AAS

V. Experimentos e Discussões

Esta seção descreve os experimentos realizados para avaliar como o chat especializado, baseado em modelos de linguagem, consegue explorar estruturas AAS por meio de comandos em linguagem natural. Os testes foram conduzidos em diferentes cenários, abrangendo diversos tipos de perguntas, modelos e estruturas. Ao final, é feita uma breve discussão dos resultados obtidos, destacando os principais pontos observados.

Para a avaliação, foram definidos dois grupos distintos, visando analisar o desempenho em contextos variados. A seguir, detalham-se as características de cada grupo:

Grupo A: Composto por 2 ativos, totalizando 6.089 linhas no arquivo JSON, sendo:

- Pepperl+Fuchs – Sensor de distância
- ABB – Transmissor de temperatura

Detalhes completos podem ser encontrados em [19].

Grupo B: Composto por 5 ativos, totalizando 18.875 linhas no arquivo JSON, incluindo:

- Schneider Electric – Motor Starter (TeSys island)
- Lenze – Servoumrichter i950 (Inversor)
- Pepperl+Fuchs – Sensor de distância
- ABB – Transmissor de temperatura
- Festo – OVEL Vacuum generator (Gerador de vácuo)

Detalhes completos podem ser encontrados em [20].

Os arquivos utilizados nos experimentos foram obtidos a partir do repositório oficial de exemplos mantido pela organização *Industrial Digital Twin Association (IDTA)* [21], que é responsável pelo desenvolvimento do AAS, e disponibiliza uma variedade de amostras de ativos no formato padronizado.

Com a finalidade de testar o chat especializado, foram elaboradas 15 perguntas divididas em três categorias, que representam diferentes níveis de complexidade:

- **Consultas diretas**: extração de informações pontuais e diretas.
  Exemplo: Qual é o ContactInfo do ativo de idShort "PeFu_267075_100078"?
- **Consultas relacionais**: exigem navegação entre múltiplos elementos.
  Exemplo: Quais os submodelos em comum entre todos os ativos?
- **Consultas interpretativas**: requerem inferência ou compreensão de contexto.
  Exemplo: Em relação a esse grupo AAS, faça um resumo do ativo de globalId: www.company.com/ids/asset/1595_3140_0191_3961

Cada pergunta foi submetida duas vezes a cada grupo, uma utilizando o modelo `GPT-4o mini` e outra com o modelo `GPT-4.1 mini`, totalizando 60 execuções.

A avaliação de cada resposta foi realizada manualmente, com base nos seguintes critérios para contabilizar a taxa de acerto:

- **0,0** – Resposta incorreta ou não informativa (incluindo respostas genéricas como "Desculpe, não posso ajudar com isso.").
- **0,5** – Resposta parcialmente correta, incompleta ou imprecisa, mas ainda útil.
- **1,0** – Resposta correta e completa, atendendo exatamente à pergunta.

A figura 8 apresenta exemplos reais obtidos durante os testes, ilustrando os três tipos de avaliação: resposta incorreta, parcialmente correta e correta.

Pergunta	Resposta	Classificação da resposta	Observação
Quais são os nomes dos submodelos do ativo com idShort "PeFu_267075_100078"?	Os nomes dos submodelos do ativo de idShort "PeFu_267075_100078" são: Nameplate, Document, Service, Identification	Correta (1)	
Qual é o ContactInfo do ativo de idShort "PeFu_267075_100078"?	O ContactInfo do ativo de idShort "PeFu_267075_100078" é: Name of Supplier: Pepperl+Fuchs Role: Manufacturer Physical Address: Country Code: DE Street: Lilienthalstr. 200 Zip: 68307 City/Town: Mannheim State/County: Baden-Württemberg Email: info@pepperl-fuchs.com URL: https://www.pepperl-fuchs.com Phone Number: +496217760 Fax: +49711 347-2190	Parcialmente Correta (0.5)	O campo "Fax" deveria estar vazio, mas a resposta apresenta um número de fax pertencente a outro ativo.
Qual é o valor do ManufacturerProductDescription do ativo de idShort "ABB_TTF_300"?	O valor do "ManufacturerProductDescription" do ativo de idShort "ABB_TTF_300" é: "Field mounted temperature transmitters".	Incorreta (0)	Na verdade, esse valor está presente na propriedade *ManufacturerProductDesignation*, e não em *ManufacturerProductDescription*, que contém um breve texto descritivo.

Fig. 8. Exemplo de perguntas e respostas

As médias das pontuações foram utilizadas para calcular as taxas de acerto por tipo de pergunta, por grupo de arquivos e

por modelo de linguagem.

A. Desempenho por Tipo de Pergunta

A Tabela I apresenta os resultados médios de desempenho, considerando todos os grupos, utilizando o modelo `GPT-4o mini` e `GPT-4.1 mini`.

TABLE I
DESEMPENHO POR TIPO DE PERGUNTA

Tipo de Pergunta	Taxa de acerto (%)	Tempo Médio (s)
Consultas diretas	84,21	2,25
Consultas relacionais	94,44	3,35
Consultas interpretativas	90,00	2,43

Os resultados demonstram que as consultas relacionais tiveram a maior taxa de acerto, ainda que com maior tempo de resposta. As consultas interpretativas apresentaram desempenho consistente, com tempo de execução equilibrado. Já as consultas diretas, embora rápidas, tiveram menor acurácia, principalmente quando utilizado o modelo *GPT-4o mini*, possivelmente devido à grande quantidade de campos na estrutura dos grupos AAS.

B. Comparativo entre Grupos

A Tabela II apresenta o desempenho médio do sistema por grupo, evidenciando o impacto da quantidade de arquivos na sua performance.

TABLE II
DESEMPENHO POR GRUPO DE AAS

Grupo	Taxa de acerto (%)	Tempo Médio (s)
Grupo A	100,00	2,16
Grupo B	77,77	3,20

Observa-se que o Grupo A, com menor volume de arquivos, obteve desempenho excelente em precisão e tempo de resposta. Já o Grupo B apresentou redução significativa na taxa de acerto, indicando que a complexidade estrutural e o volume de dados afetam negativamente a performance.

C. Comparativo entre Modelos de LLM

Para avaliar o impacto do modelo de linguagem na qualidade das respostas, todas as perguntas foram realizadas utilizando dois modelos diferentes: `GPT-4o mini` e `GPT-4.1 mini`. A tabela III apresenta os resultados obtidos.

TABLE III
COMPARATIVO ENTRE MODELOS DE LLM

Modelo	Precisão Geral (%)	Tempo Médio (s)
GPT-4o mini	82,14	2,51
GPT-4.1 mini	96,55	2,85

O modelo `GPT-4.1 mini` superou significativamente o desempenho do `GPT-4o mini`. A ligeira elevação no tempo de resposta foi compensada pelo ganho expressivo em precisão.

D. Discussão

Os resultados obtidos revelam aspectos relevantes sobre o desempenho de modelos de linguagem natural aplicados à exploração de estruturas AAS. Um dos pontos mais notáveis foi a superioridade das consultas relacionais, que apresentaram a maior taxa de acerto, mesmo exigindo navegação entre múltiplos elementos. Isso sugere que a arquitetura dos modelos avaliados lida bem com o reconhecimento de padrões entre os diferentes ativos.

Por outro lado, as consultas diretas, apesar de aparentemente mais simples, foram as que apresentaram menor taxa de acerto. Isso pode estar associado à grande diversidade de campos que há na estrutura AAS, além de que muitos campos aparecem mais de uma vez, o que dificulta a identificação de propriedades pontuais pelos modelos.

As consultas interpretativas apresentaram desempenho consistente, com alta taxa de acerto. Por se tratarem de perguntas mais subjetivas, os resultados podem variar conforme a clareza, o contexto e a formulação dos prompts. Isso reforça a importância da construção de prompts bem elaborados.

Quanto ao impacto do volume de dados, a diferença significativa entre os grupos A e B demonstra que o aumento da complexidade da estrutura afeta diretamente o desempenho, tanto em termos de precisão quanto de tempo de resposta. Esses resultados indicam, para trabalhos futuros, a importância de adotar estratégias de otimização, como a redução e a limpeza dos arquivos AAS, eliminando campos irrelevantes para a atuação da LLM.

Por fim, a comparação entre os modelos mostrou que versões mais recentes e robustas, como o `GPT-4.1 mini`, são significativamente mais capazes de lidar com tarefas interpretativas e consultas sobre estruturas maiores, mesmo com pequeno acréscimo na latência. Adicionalmente, como o sistema foi desenvolvido com a biblioteca LangChain, há flexibilidade para substituição dos modelos de linguagem de forma simples. Isso significa que, à medida que modelos mais avançados forem disponibilizados, o sistema poderá evoluir em termos de precisão e desempenho, permitindo sua longevidade tecnológica.

VI. CONCLUSÃO

Os experimentos realizados demonstraram a viabilidade da utilização do chat especializado baseado em LLMs para a exploração de estruturas AAS por meio de linguagem natural. A abordagem permite acessar informações complexas de forma simplificada, contribuindo para a redução de barreiras técnicas no uso do AAS em contextos industriais.

A análise dos resultados demonstrou que a eficácia das respostas do chat varia conforme o tipo de pergunta, a complexidade dos ativos e o modelo de linguagem utilizado. Perguntas relacionais apresentaram o melhor desempenho, enquanto perguntas diretas mostraram maior dificuldade na taxa de acerto. Perguntas interpretativas também obtiveram bons resultados, especialmente aquelas que solicitavam resumos dos ativos. Esse tipo de tarefa, como sintetizar a estrutura de um

AAS, seria consideravelmente mais difícil sem o apoio de um LLM.

A metodologia de avaliação adotada, com atribuição de escores para respostas completas, parciais e incorretas, permitiu uma análise mais refinada da qualidade das respostas. A variação de desempenho entre os grupos A e B reforça o impacto da complexidade estrutural na capacidade do sistema de navegação e inferência.

Como trabalhos futuros, propõe-se a ampliação dos testes com novos tipos de ativos, a experimentação com diferentes estratégias de recuperação de contexto e a otimização das estruturas dos arquivos AAS — por exemplo, com a remoção de campos redundantes ou irrelevantes para o modelo de linguagem. Além disso, planeja-se integrar a solução a sistemas industriais reais, de modo a avaliar seu desempenho em cenários práticos. Espera-se que essa linha de pesquisa possa contribuir para a adoção do AAS, ao tornar a interação com ativos digitais mais acessível, intuitiva e eficiente, preservando as vantagens da padronização e garantindo a interoperabilidade essencial para a Indústria 4.0.

VII. Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo incentivo e apoio, fundamentais para a realização deste trabalho.

References

[1] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for implementing the strategic initiative industrie 4.0," 2013.

[2] S. Bader, E. Barnstedt, H. Bedenbender, B. Berres, M. Billmann, B. Boss, N. Braunisch, A. Braunmandl, E. Clauer, C. Diedrich, *et al.*, "Details of the asset administration shell. part 1 -the exchange of information between partners in the value chain of industrie 4.0 (version 3.0rc02)," tech. rep., Plattform Industrie 4.0, 2022.

[3] S. R. Bader and M. Maleshkova, "The semantic asset administration shell," in *Semantic Systems. The Power of AI and Knowledge Graphs: 15th International Conference, SEMANTiCS 2019, Karlsruhe, Germany, September 9–12, 2019, Proceedings 15*, pp. 159–174, Springer, 2019.

[4] Y. Xia, Z. Xiao, N. Jazdi, and M. Weyrich, "Generation of asset administration shell with large language model agents: Toward semantic interoperability in digital twins in the context of industry 4.0," *IEEE Access*, vol. 12, pp. 84863–84877, 2024.

[5] Y. Xia, N. Jazdi, and M. Weyrich, "Automated generation of asset administration shell: a transfer learning approach with neural language model and semantic fingerprints," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–4, 2022.

[6] Eclipse AASX Package Explorer and Server, "Package explorer." https://github.com/eclipse-aaspe/package-explorer, 2025.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[8] S. Zhou, U. Alon, F. F. Xu, Z. Wang, Z. Jiang, and G. Neubig, "Docprompting: Generating code by retrieving the docs," *arXiv preprint arXiv: 2207.05987*, 2022.

[9] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023.

[10] M. Allouch, A. Azaria, and R. Azoulay, "Conversational agents: Goals, technologies, vision and challenges," *Sensors*, vol. 21, no. 24, 2021.

[11] A. Polenghi, I. Roda, M. Macchi, and A. Pozzetti, "Information as a key dimension to develop industrial asset management in manufacturing," *Journal of Quality in Maintenance Engineering*, vol. 28, pp. 567–583, 04 2021.

[12] I. Sara, K. A. K. SAPUTRA, and I. Utama, "The effects of strategic planning, human resource and asset management on economic productivity: A case study in indonesia," *The Journal of Asian Finance, Economics and Business*, vol. 8, no. 4, pp. 381–389, 2021.

[13] E. Mastroianni, J. Lancaster, B. Korkmann, A. Opdyke, and W. Beitelmal, "Mitigating infrastructure disaster losses through asset management practices in the middle east and north africa region," *International Journal of Disaster Risk Reduction*, vol. 53, p. 102011, 2021.

[14] S. Beden, Q. Cao, and A. Beckmann, "Semantic asset administration shells in industry 4.0: A survey," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 31–38, 2021.

[15] X. Ye and S. H. Hong, "Toward industry 4.0 components: Insights into and implementation of asset administration shells," *IEEE Industrial Electronics Magazine*, vol. 13, no. 1, pp. 13–25, 2019.

[16] ECMA International, "Ecma-404 the json data interchange standard." https://www.ecma-international.org/publications-and-standards/standards/ecma-404/, 2017. Acesso em: 28 abr. 2025.

[17] JSON.org, "Introducing json." https://www.json.org/json-en.html. Acesso em: 28 abr. 2025.

[18] LangChain4j, "Langchain4j." https://github.com/langchain4j/langchain4j, 2025. Acesso em: 27 abr. 2025.

[19] F. L. de Medeiros, "Grupo a." https://gist.github.com/FelipeLM1/3eaef56b50bd67344eb4b0b422cfd3d8, 2025. Acesso em: 26 jul. 2025.

[20] F. L. de Medeiros, "Grupo b." https://gist.github.com/FelipeLM1/b46a52ce719f77c39b3e77c31b678358, 2025. Acesso em: 26 jul. 2025.

[21] Industrial Digital Twin Association (IDTA), "Asset administration shell – samples." https://admin-shell-io.com/samples/, 2025. Acesso em: 27 abr. 2025.