**CENTRO UNIVERSITÁRIO SENAC**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Diogo Sarti Julio**

**Lucas Patrick Farias Figueiredo**

**Patrick Peixoto Violin**

**Modernização de Sistemas Legados com Inteligência Artificial**

**São Paulo**

**2025**

**CENTRO UNIVERSITÁRIO SENAC**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Diogo Sarti Julio**

**Lucas Patrick Farias Figueiredo**

**Patrick Peixoto Violin**

**Modernização de Sistemas Legados com Inteligência Artificial**

Trabalho de Conclusão de curso apresentado ao Centro Universitário Senac – Santo Amaro como exigência parcial para obtenção do grau de Bacharel em Sistemas da Informação.

Orientador: Prof. Jose Martinele Alves Silva

**São Paulo**

**2025**

**Novus Gateway**

**Diogo Sarti Julio**

**Lucas Patrick Farias Figueiredo**

**Patrick Peixoto Violin**

Trabalho de Conclusão de curso apresentado ao Centro Universitário Senac – Santo Amaro como exigência parcial para obtenção do grau de Bacharel em Sistemas da Informação.

**BANCA EXAMINADORA:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Jose Martinele Alves Silva – SENAC

Orientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. << Nome outro membro >> – SENAC

Membro Interno e Coorientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. << Nome membro externo>> - << Instituição Externa >>

Membro Externo

São Paulo, 23 de agosto de 2025.

**AGRADECIMENTOS**

**RESUMO**

Texto do resumo ... texto... texto

**Palavras-Chave:** Palavra-chave1, Palavra-chave2, Palavra-chave3, Palavra-chaveN.

**ABSTRACT**

Texto do abstract…

**Keywords:** Keyword1, Keyword2, Keyword3, KeywordN.

**LISTA DE FIGURAS**

Nulo

**LISTA DE QUADROS**

Nulo

**LISTA DE TABELAS**

[Tabela 1 - Cronograma 12](#_Toc206882018)

[Tabela 2 - Orçamento 12](#_Toc206882019)

**LISTA DE EQUAÇÕES**

<< Se for usar. Caso contrário, remova >>

**LISTA DE ABREVIATURAS E SIGLAS**

JSON — JavaScript Object Notation (Notação de Objeto JavaScript)

CRUD — Create, Read, Update, Delete (operações básicas em dados).

SOAP — Simple Object Access Protocol (Protocolo Simples de Acesso a Objetos)

API — Application Programming Interface (Interface de Programação de Aplicações).

IA — Inteligência Artificial

HTTP / HTTPS — Hypertext Transfer Protocol (Secure) (protocolo web).

SQL — Structured Query Language (Linguagem de Consulta Estruturada)

ML — Machine Learning (aprendizado de máquina).

XML — eXtensible Markup Language (Linguagem de Marcação Extensível)

W3C — World Wide Web Consortium (organismo que padroniza web).

REST — Representational State Transfer (estilo arquitetural de APIs).

WSDL — Web Services Description Language (descrição de serviços SOAP).

Sumário

[1 INTRODUÇÃO 11](#_Toc1052747312)

[1.1 Objetivos 12](#_Toc724454093)

[1.1.1 Objetivo Geral 12](#_Toc1398063701)

[1.1.2 Objetivos específicos 12](#_Toc1337394807)

[1.2 Delimitação do estudo 13](#_Toc960722331)

[1.3 Relevância da pesquisa 13](#_Toc2062845915)

[1.4 Metodologia 14](#_Toc29885889)

[1.5 Estrutura da documentação técnica 15](#_Toc1123093450)

[1.6 Cronograma 16](#_Toc416513953)

[1.7 Orçamento 17](#_Toc1709533272)

[2 REFERENCIAL TEÓRICO 18](#_Toc1799653252)

[3 PROPOSTA DA APLICAÇÃO 19](#_Toc1187883541)

[3.1 Descrição da aplicação 20](#_Toc2124311540)

[3.2 Modelagem dos requisitos 20](#_Toc1627258674)

[3.2.1 Requisitos funcionais 20](#_Toc495347770)

[3.2.2 Requisitos não funcionais 20](#_Toc1109510866)

[3.3 Casos de uso 20](#_Toc334363634)

[3.3.1 Diagrama de caso de uso 20](#_Toc1517983002)

[3.4 Diagramas de Classes 20](#_Toc165509694)

[3.5 Regras de negócio 21](#_Toc840008515)

[3.6 Protótipo da aplicação (Wireframe) 21](#_Toc1855258114)

[3.7 Modelagem do banco de dados 21](#_Toc1238992069)

[3.7.1 Modelo entidade relacionamento (DER) 21](#_Toc712306269)

[3.7.2 Modelo físico 21](#_Toc60662587)

[3.8 Infraestrutura da aplicação 21](#_Toc1412871036)

[4 CONSIDERAÇÕES FINAIS E SUGESTÕES DE TRABALHOS FUTUROS 22](#_Toc851143341)

[REFERÊNCIAS 23](#_Toc571815036)

[APÊNDICES << Opcional >> 24](#_Toc1999090974)

[ANEXOS << Opcional >> 25](#_Toc1792534043)

# INTRODUÇÃO

As organizações hoje dependem de trocas constantes de dados entre sistemas distintos. Esse ecossistema cresce ano a ano e segue atraindo investimento mesmo em cenários econômicos desafiadores, o que evidencia a centralidade das integrações para a operação moderna (Postman, 2023). Em termos simples: sem integrações confiáveis, aplicativos não “conversam”, relatórios não fecham e processos ficam lentos.

O problema é que uma grande parte do mundo corporativo ainda roda sistemas legados que expõem serviços no padrão SOAP, baseado em XML. SOAP é uma especificação formal do W3C para troca de mensagens estruturadas e continua presente em muitos ambientes empresariais, especialmente quando há requisitos rígidos de confiabilidade e governança (Gudgin et al., 2007; IBM, 2023). Em paralelo, fornecedores atuais estimulam o consumo via APIs REST/JSON, o que cria um “descompasso” tecnológico entre o velho e o novo (Microsoft, 2024).

Esse descompasso não é apenas técnico; ele se traduz em custo, risco e atraso. Órgãos públicos e empresas reconhecem que modernizar legados “de uma vez só” é caro e demorado; por isso, muitas iniciativas recomendam estender a vida desses sistemas por meio de camadas de API que reduzam o acoplamento e preparem uma transição gradual (U.S. DoD, 2023). Em termos leigos: é como colocar um “adaptador” inteligente entre a tomada antiga e o aparelho novo, permitindo que ambos funcionem enquanto a troca completa não acontece.

A comunidade impactada é ampla: áreas de negócio que precisam de dados em tempo real; equipes de TI que mantêm integrações frágeis; e setores regulados (como finanças e governo) onde SOAP ainda é comum. Dados de mercado mostram que profissionais, especialmente em serviços financeiros, passam muitas horas por semana apenas trabalhando com APIs, o que reforça o peso operacional dessas integrações (Postman, 2023).

Como caminho de solução, a literatura e a prática de arquitetura recomendam o uso de um gateway de APIs: uma camada única que recebe as requisições dos clientes, aplica políticas (segurança, limites, cache) e pode transformar protocolos e formatos antes de encaminhar ao back-end. Em linguagem simples, ele funciona como um “intérprete de tráfego”, simplificando o contato com sistemas antigos e novos (Microsoft TechCommunity, 2023; Microsoft Azure Architecture Center, 2024; AWS Docs, s.d.).

Para que esse intérprete seja realmente inteligente, pesquisas clássicas mostram que aprendizado de máquina ajuda a “casar” campos e estruturas entre fontes diferentes — tarefa conhecida como schema matching, tradicionalmente cara e manual (Rahm & Bernstein, 2001). Abordagens de ML reduziram esforço e aumentaram a precisão nesse mapeamento, inclusive em dados semi-estruturados como XML (Doan, Domingos & Halevy, 2003; Bernstein, Madhavan & Rahm, 2011). Em termos leigos: a máquina aprende, por exemplos, como um campo “ClienteID” de um lado corresponde a “CustomerCode” do outro, evitando retrabalho humano e erros

## Objetivos

Desenvolver e validar soluções baseadas em Inteligência Artificial que apoiem a modernização de sistemas legados, focando na conversão de APIs antigas para formatos modernos.

### Objetivo Geral

Desenvolver um gateway de APIs baseado em IA que intercepte requisições JSON, transforme-as em SOAP/XML para requisições para sistemas legados e responda para clientes modernos em formato JSON, utilizando aprendizado de máquina para automatizar e melhorar o processo de mapeamento entre os schemas. Este gateway será gerenciado através de um portal web para o usuário, através dele será feita a publicação e configuração da API.

### Objetivos específicos

Considerando que este estudo não busca substituir ferramentas consolidadas de integração ou migração, mas demonstrar a viabilidade do uso de IA em cenários controlados, os objetivos específicos desdobram-se em passos incrementais e verificáveis. O foco permanece em conversões SOAP/XML para JSON, sempre com ênfase em clareza metodológica e reprodutibilidade. Assim, os objetivos seguintes organizam o caminho de desenvolvimento, validação e documentação dos protótipos.

* Projetar um gateway inteligente capaz de interceptar requisições SOAP/XML e convertê-las em respostas JSON, utilizando técnicas de aprendizado de máquina para sugerir mapeamentos dinâmicos.
* **Coletar e organizar** pares SOAP-JSON (públicos ou sintéticos) a fim de compor um dataset de treinamento representativo.
* **Implementar** um conversor inicial baseado em regras, evoluindo posteriormente para modelos de ML que auxiliem na correspondência de esquemas e otimização do processo.
* **Treinar e validar** modelos de aprendizado de máquina para mapeamento de campos e detecção de inconsistências em schemas não padronizados.
* **Integrar** o modelo em uma API, disponibilizando processamento de requisições SOAP em tempo real.
* **Construir protótipos funcionais com interfaces de demonstração**, possibilitando testes em cenários práticos (ex.: APIs públicas SOAP convertidas em REST/JSON).
* Avaliar a eficácia e o desempenho das soluções em diferentes cenários, medindo precisão, robustez e eficiência.
* **Explorar** o uso de modelos generativos para propor melhorias, compressão de respostas e documentação automatizada.
* **Documentar** os resultados obtidos e apresentar recomendações para evoluções futuras do sistema.

## Delimitação do estudo

O projeto não pretende substituir integralmente ferramentas consolidadas de integração de sistemas, mas sim propor protótipos acadêmicos que demonstrem a viabilidade da aplicação de IA nesses contextos. O escopo está limitado a:

* **Conversão SOAP/XML → JSON** para serviços **simples**, com foco em **mapeamento de campos** e **conversão semântica** consumível por clientes modernos.
* Protótipos em ambiente controlado, sem a necessidade de integração em sistemas de missão crítica.
* **Sem cobertura de tópicos avançados de segurança corporativa** (por exemplo, gestão de identidades em larga escala e auditoria normativa completa), além do estritamente necessário aos protótipos.

## Relevância da pesquisa

A relevância desta investigação decorre de seu caráter formativo e demonstrativo: ao trabalhar com protótipos em ambiente controlado, o estudo produz evidências empíricas sobre como técnicas de IA podem auxiliar na modernização de legados sem interromper operações críticas. O recorte adotado permite discutir ganhos, limitações e riscos com base em resultados mensuráveis, oferecendo insumos tanto para a prática profissional quanto para novas pesquisas.

* Endereçar um dos maiores gargalos em Sistemas de Informação: a modernização de legados.
* Oferece evidências experimentais sobre ganhos e limites do uso de aprendizado de máquina no schema matching de mensagens XML para JSON.
* Explorar o uso de IA aplicada à engenharia de software, área em expansão e com crescente demanda no mercado.
* Contribuir com soluções que podem ser estendidas para empresas reais, oferecendo ganho em eficiência, redução de custos e continuidade de negócios.
* Fornecer material acadêmico relevante para futuras pesquisas em modernização, interoperabilidade e automação inteligente. Disponibiliza Artefatos reprodutíveis que podem ser reutilizados por equipes técnicas e por pesquisas futuras.

## Metodologia

Dado o objetivo de comprovar viabilidade em escopo limitado, a metodologia privilegia uma abordagem aplicada, com etapas de revisão, modelagem, implementação incremental e experimentação controlada. Em vez de perseguir abrangência total, opta-se por desenhos enxutos que favoreçam comparações, medição de resultados e reprodutibilidade. Essa estratégia alinha-se ao propósito acadêmico do trabalho e à necessidade de avaliar, de forma transparente, onde as técnicas testadas agregam valor e onde encontram barreiras.

A estrutura proposta organiza o texto de modo a conduzir o leitor do contexto à evidência: inicia-se pela introdução do problema e do recorte; avança-se para o embasamento teórico que sustenta as escolhas; detalham-se os protótipos e o desenho experimental; e, por fim, apresentam-se resultados, limitações e caminhos futuros. Essa sequência facilita a rastreabilidade entre decisões de projeto, execução dos experimentos e conclusões, mantendo a coerência com o escopo restrito e o caráter demonstrativo do estudo.

* Revisão bibliográfica sobre modernização de sistemas, APIs e uso de IA.
* Modelagem conceitual dos dois protótipos propostos.
* Implementação incremental utilizando Python, framework web FastAPI, bibliotecas de IA (scikit-learn, Hugging Face), ferramentas de integração de dados (pandas, SQLAlchemy), bibliotecas XML/JSON, scikit-learn e ferramentas para experimentos reprodutíveis.
* Construção de baseline por regras; geração/coleta de pares SOAP–JSON; treinamento e validação de modelos; integração em um serviço web; experimentos controlados com métricas.
* Análise dos resultados quanto à precisão dos mapeamentos, desempenho e aplicabilidade prática.

## Estrutura da documentação técnica

A estrutura proposta organiza o texto de modo a conduzir o leitor do contexto à evidência: inicia-se pela introdução do problema e do recorte; avança-se para o embasamento teórico que sustenta as escolhas; detalham-se os protótipos e o desenho experimental; e, por fim, apresentam-se resultados, limitações e caminhos futuros. Essa sequência facilita a rastreabilidade entre decisões de projeto, execução dos experimentos e conclusões, mantendo a coerência com o escopo restrito e o caráter demonstrativo do estudo.

* Capítulo 1: Introdução – contextualização, objetivos, relevância e metodologia.
* Capítulo 2: Referencial Teórico – conceitos de modernização de sistemas, APIs, IA, gateways e schema matching com ML.
* Capítulo 3: Proposta da Aplicação – detalhamento dos protótipos, requisitos, diagramas e modelagens.
* Capítulo 4: Experimentos e Resultados – desenho experimental, métricas e análise crítica.
* Capítulo 5: Considerações Finais – análise dos resultados, limitações e sugestões de trabalhos futuros.

## Cronograma

* Cronograma do Planejamento do Produto

A graph with a grid and a graph

AI-generated content may be incorrect.

* Cronograma do Desenvolvimento do Produto

A graph with multiple rows of squares

AI-generated content may be incorrect.

## Orçamento

Este orçamento considera um projeto acadêmico de 11 meses com 3 analistas e foco em protótipos em ambiente controlado. Além da mão de obra, foram incluídos custos de nuvem/APIs, internet, energia elétrica (estimada para estações de 0,18 kW por 8 h/dia, 22 dias/mês, mais overhead de rede), licenças/DevOps, coworking, materiais de consumo, impressão/encadernação, deslocamentos locais, material bibliográfico e depreciação de equipamentos. Aplica-se contingência de 10% sobre os itens não laborais.



# REFERENCIAL TEÓRICO

# PROPOSTA DA APLICAÇÃO

## Descrição da aplicação

## Modelagem dos requisitos

### Requisitos funcionais

### Requisitos não funcionais

## Casos de uso

### Diagrama de caso de uso

## Diagramas de Classes

## Regras de negócio

## Protótipo da aplicação (Wireframe)

## Modelagem do banco de dados

### Modelo entidade relacionamento (DER)

### Modelo físico

## Infraestrutura da aplicação

# CONSIDERAÇÕES FINAIS E SUGESTÕES DE TRABALHOS FUTUROS

REFERÊNCIAS

* POSTMAN. 2023 State of the API Report. [S.l.]: Postman, 2023.
* GUDGIN, Martin; HADLEY, Marc; MENDELSOHN, Noah; MOREAU, Jean-Jacques; NIELSEN, Henrik F. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C Recommendation. [S.l.]: World Wide Web Consortium, 2007.
* IBM. Understanding Web Services (InfoSphere MDM Documentation). [S.l.]: IBM, 2023.
* MICROSOFT. Increasing business value by integrating SOAP legacy assets with Azure Logic Apps using Azure API Management. Microsoft TechCommunity, 2024.
* MICROSOFT. API gateways — Azure Architecture Center. Microsoft Learn, 2024.
* U.S. DEPARTMENT OF DEFENSE. DoD Approves Software Modernization Implementation Plan. Press release, 11 Apr. 2023.
* AMAZON WEB SERVICES. Amazon API Gateway — Developer Guide. [S.l.]: AWS, s.d.
* RAHM, Erhard; BERNSTEIN, Philip A. A survey of approaches to automatic schema matching. The VLDB Journal, v. 10, p. 334–350, 2001.
* DOAN, AnHai; DOMINGOS, Pedro; HALEVY, Alon. Learning to Match the Schemas of Data Sources: A Multistrategy Approach. Machine Learning, v. 50, p. 279–301, 2003.
* BERNSTEIN, Philip A.; MADHAVAN, Jayant; RAHM, Erhard. Generic Schema Matching, Ten Years Later. PVLDB, v. 4, n. 11, p. 695–701, 2011.

APÊNDICES << Opcional >>

ANEXOS << Opcional >>