DOCUMENTO DE ARQUITETURA DE SOFTWARE

Projeto: Fitness Advisor — Sistema de Consultoria com Inteligência Artificial

Autores: Iago Carvalho Souto, João Augusto Dias, João Paulo Daré, Lucas Pereira, Renan Augusto

Revisor Técnico: _____

Versão: 1.0 | Status: Confidencial

Data de Emissão: Julho de 2025

SUMÁRIO

1. INTRODUÇÃO

1.1 Finalidade

Este Documento de Arquitetura de Software tem como finalidade fornecer uma visão abrangente, consistente e detalhada do **Fitness Advisor**, um sistema distribuído baseado em agentes de Inteligência Artificial para consultoria personalizada em fitness e nutrição.

Ele descreve as principais decisões de projeto, critérios adotados para a arquitetura, definições técnicas, diagramas de suporte, além de mapear as ameaças e medidas de mitigação, visando garantir **confidencialidade, integridade, disponibilidade e auditabilidade** dos dados dos usuários.

Este documento também servirá como **referência técnica** para desenvolvedores, arquitetos de software, equipe de segurança, DevSecOps, gestores de produto e demais stakeholders, garantindo alinhamento entre as equipes durante o desenvolvimento, implantação e manutenção evolutiva do sistema.

1.2 Escopo

Este Documento de Arquitetura de Software aplica-se integralmente ao **Fitness Advisor**, um sistema distribuído de consultoria fitness com Inteligência Artificial local, projetado para oferecer **planos personalizados de treino e nutrição** a partir de dados fornecidos pelos usuários.

O escopo abrange a definição da arquitetura lógica, física e de implantação, descreve os principais fluxos de dados, casos de uso, requisitos e restrições técnicas que impactam diretamente na estrutura do sistema, além de mapear e tratar ameaças de segurança identificadas por meio da metodologia **STRIDE**.

Inclui também diretrizes de qualidade de software, dimensionamento, performance esperada e boas práticas de segurança da informação, visando assegurar que o sistema atenda aos objetivos de negócio com **resiliência, escalabilidade e confidencialidade**.

1.3 Definições, Acrônimos e Abreviações

Sigla	Descrição
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege — metodologia de modelagem de ameaças desenvolvida pela Microsoft.
DFD	Data Flow Diagram — Diagrama de Fluxo de Dados utilizado para representar visualmente os fluxos de informação no sistema.
IA	Inteligência Artificial
QoS	Quality of Service — conjunto de atributos de qualidade como desempenho, disponibilidade, escalabilidade e segurança.
OWASP	Open Worldwide Application Security Project — comunidade de referência em segurança de aplicações web.
JWT	JSON Web Token — padrão aberto (RFC 7519) para autenticação e troca segura de informações.
API	Application Programming Interface — interface que permite comunicação entre sistemas e componentes.

1.4 Referências

• OWASP Top 10
• STRIDE Threat Modeling
• FastAPI Security

2. VISÃO INICIAL PRÉ-MODELAGEM DE AMEAÇAS

Antes da implementação das medidas de mitigação, o sistema Fitness Advisor apresenta uma série de vulnerabilidades inerentes à sua arquitetura e funcionalidades. O ambiente inicial do sistema é caracterizado por:

- **Falta de controle de autenticação robusto:** Usuários e agentes do sistema podem ser alvo de ataques de spoofing, devido à ausência de mecanismos rigorosos para verificação de identidade.

- **Comunicação não segura entre componentes:** Dados sensíveis podem trafegar sem criptografia adequada, abrindo brechas para interceptação e manipulação (tampering).

- **Ausência de registros (logs) confiáveis:** Dificuldade para rastrear ações realizadas no sistema, prejudicando a detecção de repúdio e responsabilização.

- **Superfície de ataque aberta a vazamento de informações:** Sem políticas de controle de acesso adequadas, informações pessoais e resultados das análises podem ser expostos indevidamente.

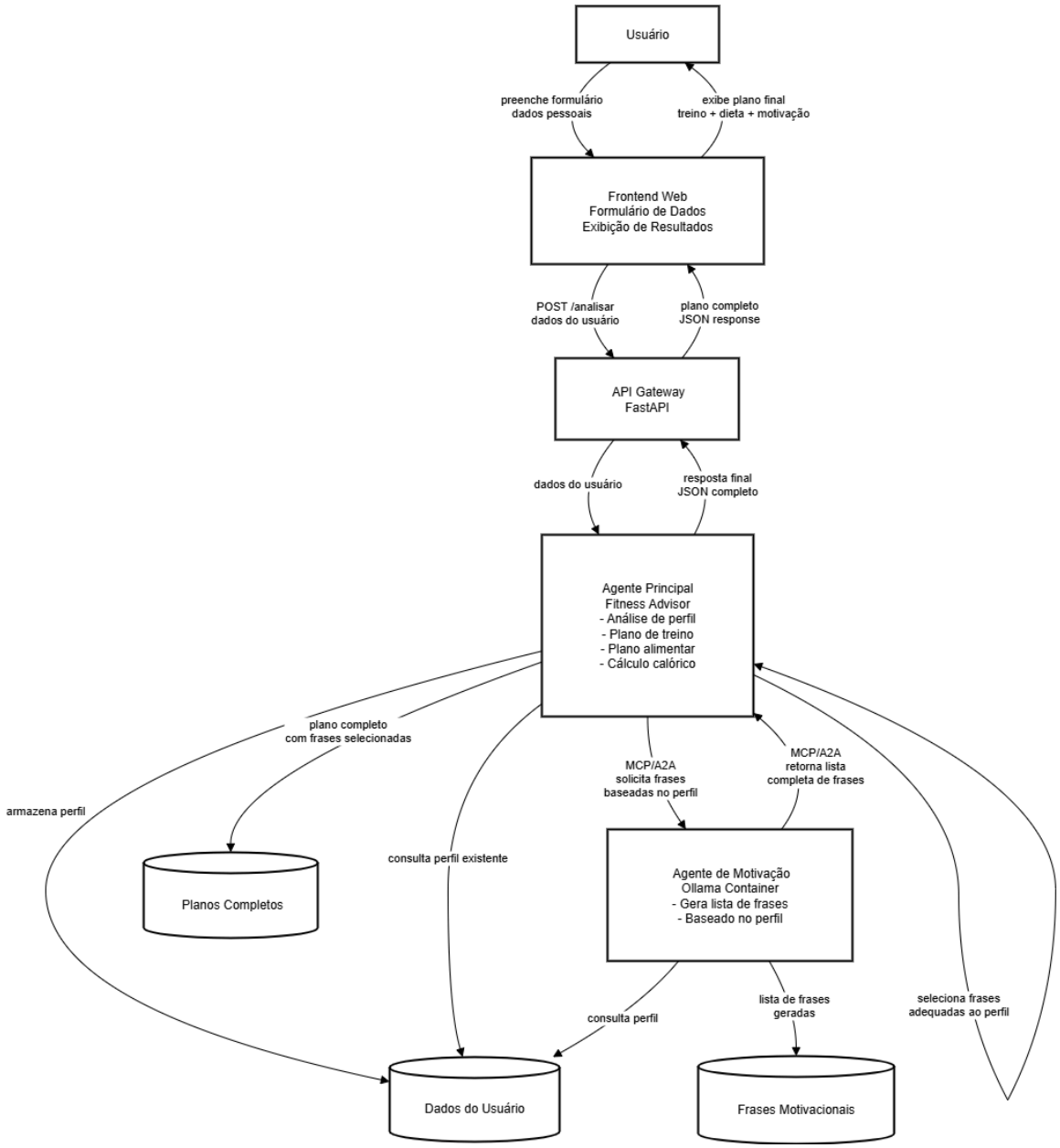
- **Falta de mecanismos de proteção contra negação de serviço:** O sistema está suscetível a ataques que visam indisponibilizá-lo.

- **Privilégios excessivos sem controle:** Possibilidade de escalonamento de privilégios devido a controles inadequados.

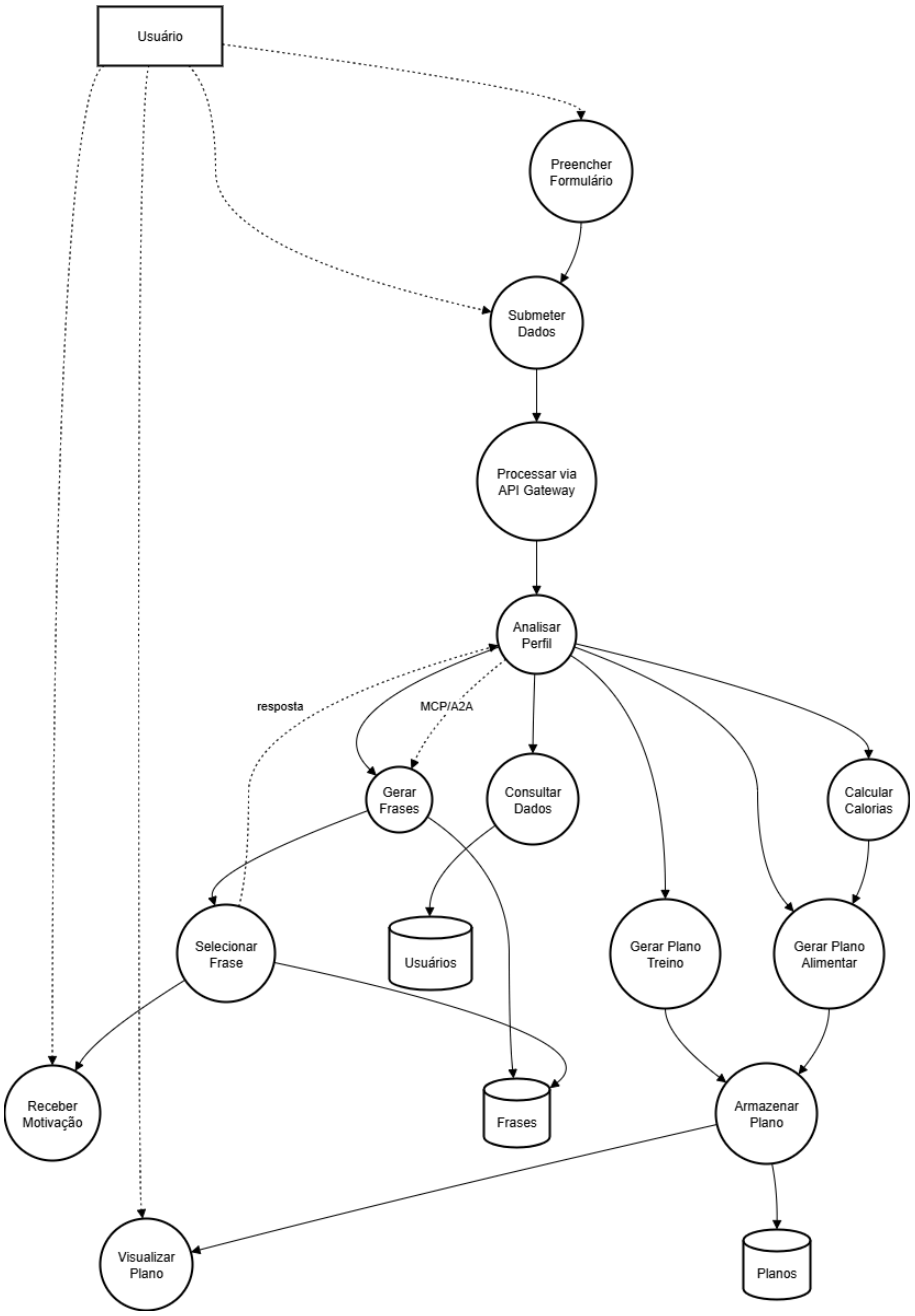
Esse cenário inicial demonstra a necessidade de uma modelagem de ameaças detalhada para identificar, classificar e priorizar as vulnerabilidades, permitindo a definição de estratégias eficazes de mitigação.

3. REPRESENTAÇÃO ARQUITETURAL

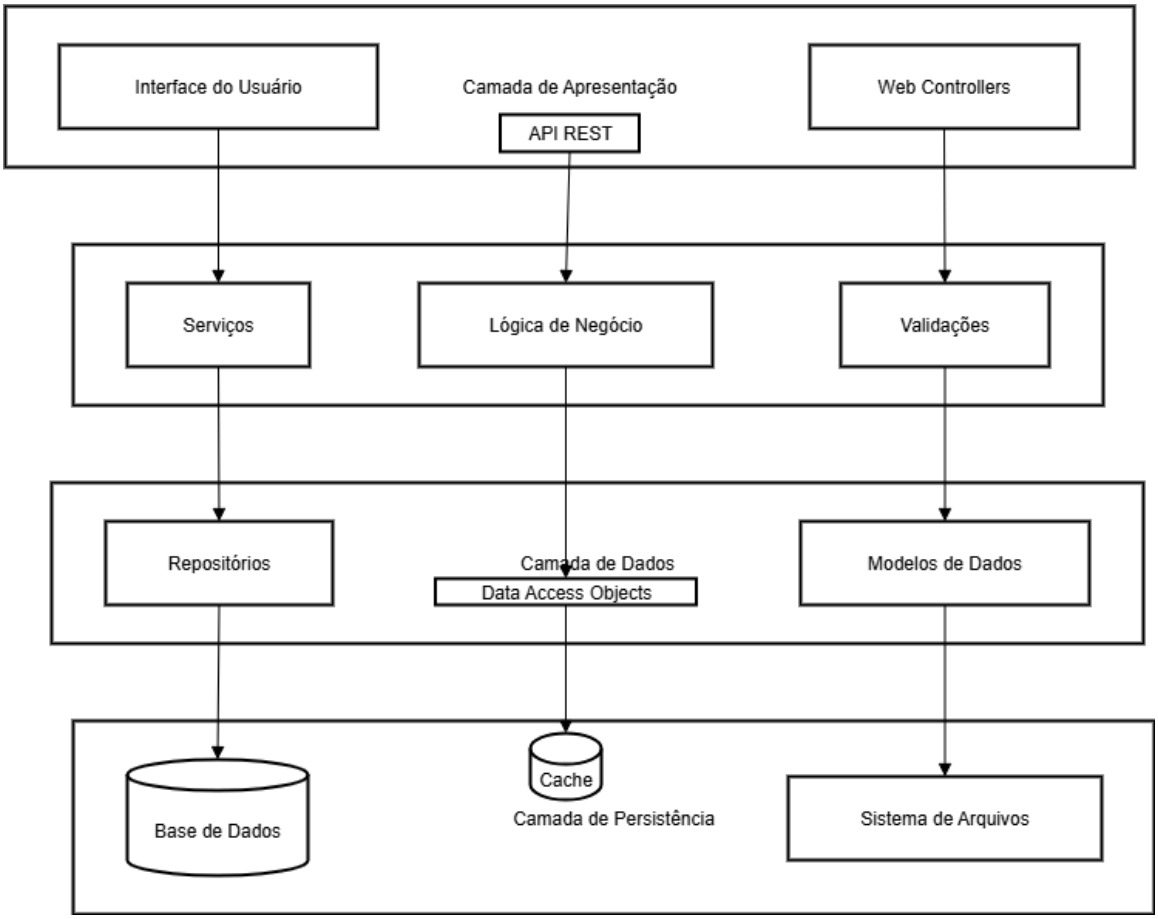
3.1 Diagrama de Fluxo de Dados



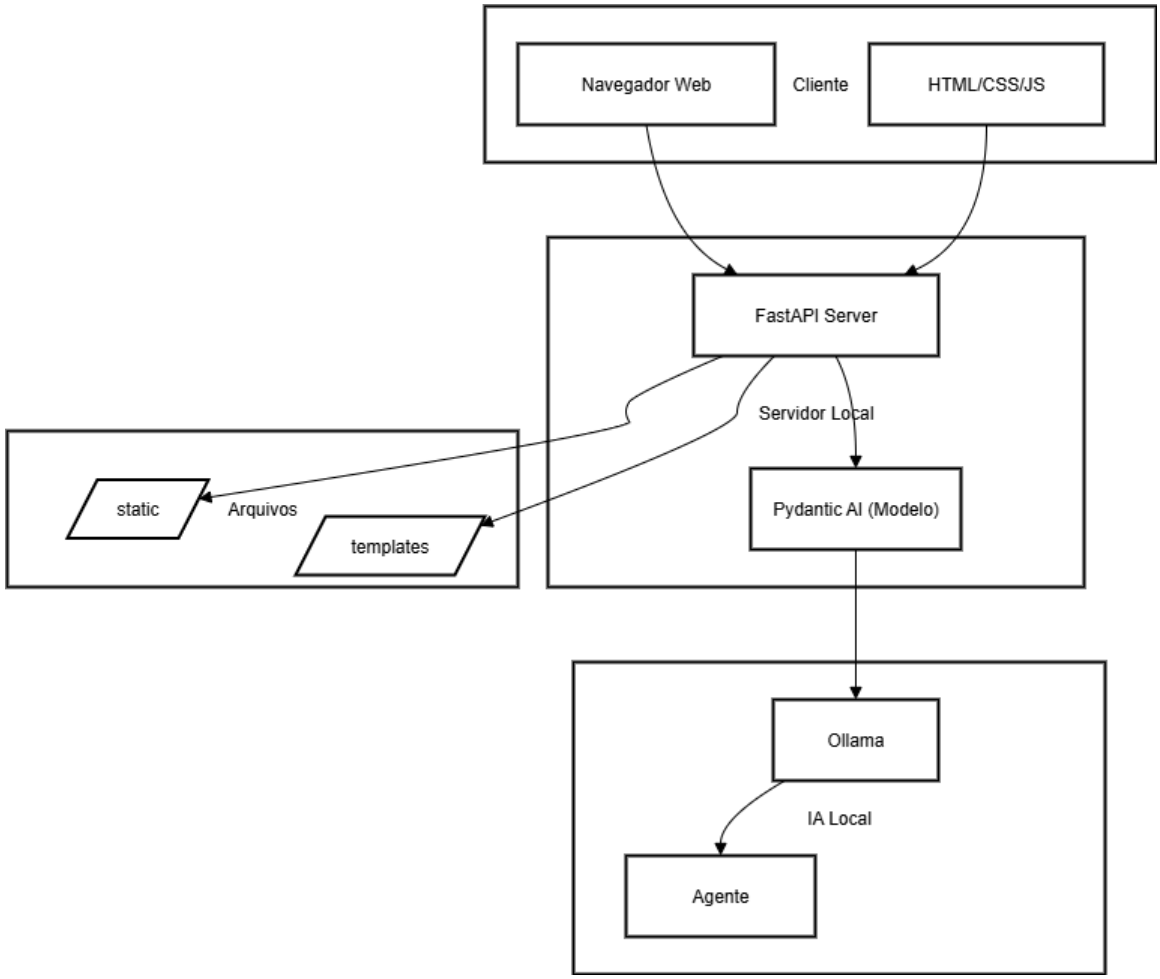
3.2 Visão de Casos de Uso



3.3 Visão Lógica



3.4 Visão de Implantação



4. REQUISITOS E RESTRIÇÕES ARQUITETURAIS

Requisitos funcionais e não funcionais, restrições técnicas e regulatórias.

4.1 Requisitos Funcionais

- Permitir que o usuário preencha um formulário com dados pessoais e informações fitness.

- Processar os dados do formulário e gerar uma ficha de consultoria personalizada.
- Exibir a ficha gerada para o usuário.
- Permitir que o usuário baixe ou copie a ficha gerada.
- Validar os dados inseridos no formulário, garantindo campos obrigatórios e formatos corretos.

4.2 Requisitos Não Funcionais

- Resposta rápida na geração da ficha, com tempo inferior a 3 segundos.
- Interface simples e intuitiva para o preenchimento do formulário.
- Compatibilidade com navegadores modernos, como Chrome, Firefox e Edge.
- Proteção dos dados durante a transmissão utilizando HTTPS.
- Estabilidade do sistema, evitando travamentos e falhas.

4.3 Restrições Técnicas

- Utilização da linguagem Python e framework FastAPI para desenvolvimento.
- Formulário simples implementado via frontend ou API que recebe JSON.
- Ausência de armazenamento persistente dos dados do usuário (processamento em tempo real).
- Implantação em ambiente local ou servidor com recursos básicos.
- Geração da ficha em formato texto ou PDF no momento da requisição.

4.4 Restrições Regulatórias

- Dados do usuário não armazenados permanentemente, garantindo privacidade.
- Transparência quanto ao uso dos dados: utilizados apenas para geração da ficha.

- Proteção dos dados pessoais durante a comunicação, respeitando boas práticas de segurança.

- O sistema não realiza diagnósticos médicos, oferecendo apenas sugestões baseadas nas informações fornecidas.

5. ANÁLISE DE AMEAÇAS (STRIDE)

Tabela detalhada de ameaças por componente.

Descrição	**Categoria STRIDE**	**Elemento afetado**	**Impacto potencial**	**Probabilidade**	**Mitigações sugeridas**
Manipulação dos dados do formulário antes do envio	Tampering	Frontend	Dados incorretos processados	Média	Validação no backend
Envio de requisições automatizadas (bots)	Denial of Service	API	Queda do serviço	Alta	Rate limiting, CAPTCHA
Dados interceptados em trânsito	Information Disclosure	API	Vazamento de dados pessoais	Média	Uso de HTTPS
Prompt injection para IA	Tampering	Service Layer/IA	Respostas inadequadas ou ofensivas	Média	Sanitização de entrada, validação de prompts
Respostas da IA com dados sensíveis	Information Disclosure	Service Layer/IA	Vazamento de informações	Baixa	Revisão de prompts, filtros de saída

Usuário nega ter enviado dados	Repudiation	API	Dificuldade de auditoria	Baixa	Logs de requisições
Flood de requisições para sobrecarregar IA	Denial of Service	Service Layer/IA	Lentidão ou indisponibilidade	Alta	Rate limiting, monitoramento

5.1 Priorização das Ameaças (Matriz de Risco)

Ameaça	**Impacto**	**Probabilidade**	**Risco (Impacto x Probabilidade)**	**Prioridade**
Envio de requisições automatizadas (bots)	Alto (15)	Alta (15)	225	1
Flood de requisições para IA	Alto (15)	Alta (15)	225	1
Dados interceptados em trânsito	Médio (10)	Média (10)	100	2
Manipulação de dados do formulário	Médio (10)	Média (10)	100	2
Prompt injection para IA	Médio (10)	Média (10)	100	2

Respostas da IA com dados sensíveis	Baixo (5)	Baixa (5)	25	3
Usuário nega ter enviado dados	Baixo (5)	Baixa (5)	25	3

6. PLANO DE MITIGAÇÃO

Ações, prazos e controles sugeridos.

Validação rigorosa dos dados no backend: Utilização do Pydantic para garantir que todas as entradas recebidas estejam dentro do formato esperado, evitando dados inválidos ou maliciosos.

Controle de taxa (Rate limiting): Implementação de limites de requisições por usuário ou IP para prevenir ataques de negação de serviço (DoS) e uso abusivo do sistema, podendo ser aplicado via middleware.

Uso obrigatório de HTTPS em produção: Garantir que toda a comunicação entre cliente e servidor seja criptografada para proteger os dados transmitidos contra interceptação.

Sanitização de entradas e prompts: Limpeza e verificação das informações recebidas antes de enviar para sistemas de IA, prevenindo injeções ou comandos maliciosos.

Registro de logs de requisições: Manter logs detalhados das interações para auditoria, rastreamento de problemas e análise de segurança.

Configuração restrita de CORS: Permitir acesso apenas a origens confiáveis para evitar ataques Cross-Origin e proteger os dados do sistema.

Monitoramento contínuo do uso: Acompanhar padrões de acesso para identificar e reagir a comportamentos anômalos ou abusivos, garantindo estabilidade e segurança.

Revisão e atualização contínua: O DFD e a lista de ameaças devem ser revisados a cada nova funcionalidade. Novas ameaças e mitigações devem ser documentadas conforme o sistema evolui.

7. VISÃO FINAL APÓS IMPLEMENTAÇÃO

Descrição da postura de segurança após mitigação.

Este documento apresenta a modelagem arquitetural e a análise de ameaças de um sistema distribuído baseado em agentes de inteligência artificial, com foco na segurança e na integridade dos dados em trânsito. O sistema é composto por uma aplicação web com frontend em HTML/JavaScript, uma API desenvolvida em FastAPI, uma camada de serviços responsável pela lógica de negócio e integração com os agentes de IA (via Ollama e pydantic-ai), e fluxos de comunicação estruturados em requisições HTTP.

A ausência de armazenamento persistente reforça a importância da proteção dos dados temporários que transitam entre as camadas do sistema. Por meio da construção de um Diagrama de Fluxo de Dados (DFD), foi possível identificar as principais interfaces do sistema, seus limites de confiança e os caminhos críticos dos dados desde a entrada do usuário até a resposta gerada pela IA.

Com base na metodologia STRIDE, foram mapeadas ameaças potenciais, categorizadas de acordo com os elementos do sistema, e propostas medidas de mitigação para cada caso. A priorização dessas ameaças foi realizada por meio de uma matriz de risco que leva em conta impacto e probabilidade, fornecendo direcionamento estratégico para ações corretivas e preventivas.

As principais ameaças identificadas incluem ataques de negação de serviço (DoS), manipulação de entrada (tampering), interceptação de dados e injeção de prompts maliciosos. Entre as principais mitigações sugeridas estão o uso de HTTPS, validação de entrada com Pydantic, sanitização de prompts, controle de CORS e aplicação de rate limiting.

O trabalho destaca a necessidade de atualização contínua da análise de segurança conforme o sistema evolui, reforçando uma abordagem proativa frente às ameaças cibernéticas em ambientes distribuídos que envolvem processamento com IA.

8. DIMENSIONAMENTO E PERFORMANCE

Volume estimado de usuários, requisições e SLA de resposta.

- O sistema é projetado para uso básico com poucos usuários simultâneos.
- Espera-se que o tempo de resposta para gerar a ficha seja rápido, idealmente abaixo de 3 segundos.
- Não há estimativa precisa de volume de usuários ou requisições no momento.
- O sistema deve ser estável e funcionar corretamente mesmo com variações pequenas no uso.
- Futuramente, pode ser necessário ajustar o sistema conforme o aumento da demanda.

9. QUALIDADE (QOS)

Escalabilidade, disponibilidade, segurança.

- O sistema deve ser estável e funcionar sem falhas na maior parte do tempo.
- A segurança deve garantir a proteção dos dados durante a comunicação.
- A escalabilidade será considerada conforme a necessidade, mas não é prioridade inicial.

10. CONCLUSÃO

Sumariza decisões arquiteturais, riscos mitigados e próximos passos.

Este sistema adota uma arquitetura simples focada em processar formulários e gerar fichas rapidamente, sem armazenamento persistente. As decisões priorizam facilidade de uso, segurança básica (validação de dados, HTTPS) e estabilidade. Os principais riscos, como

entrada de dados inválidos e abuso do sistema, serão mitigados com validação no backend, controle de acesso e monitoramento. Como próximos passos, recomenda-se implementar rate limiting, monitoramento mais robusto e avaliar a escalabilidade conforme o crescimento dos usuários.