



Projeto Final - Interface Humano-Computador (IHC) 2025_2

1. Identificação do Projeto

- **Nome do Projeto:** NetSafe AI
 - **Grupo:** Equipe Aegis
 - **Integrantes:**
 - Felipe Barcelos de Carvalho (Líder NetSafe / IA) - 22350044
 - Marcos Vinícius (Líder Aegis / Full-Stack) - 22352865
 - João Marcelo (QA / Docs) - 22350653
 - João Luiz (QA / Docs) - 22353899
 - Eduardo Uchoa (Back-end) - 22353207
 - **Data de Início:** 07/11/2025
 - **Data de Entrega:** 05/12/2025
-

2. Tema e Público-Alvo

- **Domínio de Aplicação:** Cibersegurança / Monitoramento de Ameaças e Educação Tecnológica.
 - **Descrição do Público-Alvo:**
 - **Primário:** Analistas de SOC (Security Operations Center) e estudantes de cibersegurança que necessitam de dados técnicos rápidos sobre vulnerabilidades (CVEs).
 - **Secundário:** Gestores de TI que precisam de uma visão macroscópica (dashboards) e explicações simplificadas sobre riscos para tomada de decisão.
 - **Terciário:** Pessoas comuns que utilizam tecnologia no dia a dia e querem saber mais sobre a área de segurança digital
 - **Justificativa da Escolha:** A área de segurança sofre com a fragmentação de dados e a complexidade técnica. NetSafe AI justifica-se pela necessidade de acessibilidade e eficiência: ele remove a barreira de navegação complexa, permitindo que o usuário obtenha dados críticos através de linguagem natural, atendendo tanto o especialista (que quer rapidez) quanto o iniciante (que precisa de explicação).
-

3. Pesquisa e Análise de Requisitos

- **Método de Pesquisa Utilizado:**
 - Análise de outros Protótipos e Sites que já estão no mercado.
 - Brainstorming focado em **Dores do Usuário**: "Dificuldade em entender siglas técnicas" e "Tempo gasto procurando informações em menus".
- **Principais Necessidades Identificadas:**
 - Visualização imediata de tendências de ataques.
 - Necessidade de um "glossário ativo" para termos técnicos.
 - Informações de forma acessível
 -
- **Expectativas do usuário:**
 - ChatBot que consiga explicar tudo sobre segurança digital e sobre o Dashboard
 - Obter respostas em menos de 3 cliques ou através de uma pergunta direta. Interface com modo escuro (padrão da indústria) para evitar fadiga ocular.
 -
- **Referências:**
 - **GOOGLE.** *Google AI Studio Documentation: Gemini API*. Disponível em: <https://ai.google.dev/docs>. Acesso em: dez. 2025. (Documentação da IA utilizada).

- GRINBERG, Miguel. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2018. (Referência técnica sobre o framework usado no Back-end).
 - BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana da. *Interação humano-computador*. Rio de Janeiro: Elsevier, 2010.
-

4. Definição de Usuários e Atividades

- **Personas Criadas:**
 1. **David (O Aprendiz):** 20 anos, estagiário. **Objetivo:** Aprender sobre novas vulnerabilidades. **Limitação:** Não conhece todas as siglas técnicas. Precisa do NetSafe para explicar "O que é um Rootkit?"
 2. **Julian (O experiente):** 45 anos, analista. **Objetivo:** Relatórios rápidos. **Limitação:** Pouco tempo disponível. Usa o NetSafe para perguntar "Qual o setor mais atacado hoje?" sem abrir vários gráficos.
 - **Atividades Planejadas:**
 1. **Consultar estatísticas de CVEs no Dashboard.**
 2. **Utilizar o Chatbot para tirar dúvidas conceituais (ex: "Como prevenir Ransomware?").**
 3. **Alternar entre Tema Claro e Escuro para conforto visual.**
-

5. Wireframes e Validação UX

- **Ferramenta Utilizada:** Figma e Prototipagem de Alta Fidelidade (HTML/CSS direto).
 - **Método de Validação:** Walkthrough Cognitivo (Simulação de uso das personas).
 - **Feedback Recebido:** Usuários relataram que, embora os gráficos fossem úteis, sentiam falta de um guia para interpretar os dados.
 - **Ajustes Realizados:**
 - Implementação do NetSafe AI como botão flutuante constante.
 - Adição de "Chips" (Sugestões de Perguntas) dentro do chat para reduzir a carga cognitiva (Lei de Hick), facilitando o início da interação.
-

6. Implementação Técnica

- **Tecnologias Utilizadas:**
 1. **Frontend:** HTML5, CSS3 (com Variáveis para Temas), JavaScript (Vanilla).
 2. **Backend (IA):** Python 3.12, Flask, Google Gemini API (Modelos 2.0 Flash).
 3. **Técnica de IA:** RAG (Retrieval-Augmented Generation) para garantir respostas baseadas nos dados do projeto ([dados_seguranca.json](#)).
 - **Design Responsivo:** Sim. (O Chatbot e o Dashboard se adaptam a telas menores via Flexbox e unidades relativas).
 - **LINK do Protótipo Funcional:** [LINK](#)
 - **Principais Funcionalidades Implementadas:**
 1. **Chatbot Inteligente:** Responde sobre segurança e sobre o próprio projeto.
 2. **Sincronização de Tema:** O Chatbot detecta se o site está em modo Claro ou Escuro e se adapta automaticamente.
 3. **Guardrails (Segurança da IA):** O bot foi programado para recusar assuntos fora de contexto (ex: culinária), mantendo o foco profissional.
-

7. Integração e Testes

- **Integração HTML/CSS/JS:** O projeto segue uma arquitetura modular. O NetSafe possui seus próprios arquivos ([chatbot.js](#), [chatbot.css](#)) que são importados no [index.html](#) principal. Isso permite que a IA seja "plugada" em qualquer página do sistema sem quebrar o layout existente.
 - **Testes de Usabilidade:**
 - *Teste de Eficiência:* Comparação do tempo para descobrir "O que é Phishing" via busca no Google vs. Pergunta no NetSafe. O NetSafe reduziu o tempo da tarefa em cerca de 70% o tempo de pesquisa de uma pessoa que iria sair do Dashboard e pesquisar no Google.
 - **Testes de Acessibilidade:**
 - **Contraste:** O sistema garante alto contraste em ambos os temas (Claro/Escuro), facilitando a leitura.
 - **Navegação:** O chat pode ser aberto e fechado, e o foco do teclado é gerenciado para o campo de input.
-

8. Documentação Final

- **Resumo das Decisões de Design:** Optamos por um design "Dark Mode" nativo para alinhar com a estética de ferramentas de cibersegurança (hacker aesthetic). O mascote do NetSafe foi desenhado para ser amigável, humanizando a interação técnica.

Dificuldades Encontradas e Soluções:

- *Dificuldade:* A IA alucinava respostas genéricas.
- *Solução:* Implementação de RAG (Retrieval-Augmented Generation), forçando o modelo a ler o arquivo JSON do projeto antes de responder.
- *Dificuldade:* Integração do Python com GitHub Pages (Estático).
- *Solução:* Separação da arquitetura em Frontend (Pages) e Backend Local/API, garantindo segurança das chaves de API.

Considerações Finais: O NetSafe elevou o nível do Projeto Aegis de um simples painel visual para uma **plataforma inteligente**. Ele atende aos requisitos de IHC ao prover flexibilidade, consistência e ajuda contextual, tornando a segurança digital mais acessível para todos.

Declaração de Autoria: Declaramos que este projeto foi desenvolvido integralmente pelo grupo, sem plágio ou uso indevido de conteúdo de terceiros.