

Universidade Federal de Pernambuco :: Centro de Informática  
Sistemas de Informação :: [IF1006] Transformação Digital com IA  
Prof. Vinicius Cardoso Garcia

# Solicitação de Proposta Didática (RFP)

## 1. Introdução

Este RFP pretende guiar as equipes de alunos na execução de um projeto estratégico intensivo em IA generativa, utilizando a metodologia **Sinfonia** como base. A proposta deve abordar um problema de negócio relevante, aplicando técnicas de engenharia de software, práticas de IA e abordagens éticas. O resultado esperado é uma solução digital funcional e bem documentada, alinhada aos critérios de qualidade estabelecidos.

O projeto permitirá que os alunos vivenciem as etapas da metodologia **Sinfonia**, desde a imersão no problema até a validação da entrega do produto final, aplicando as boas práticas discutidas em sala de aula.

---

## 2. Objetivos

1. Desenvolver uma solução digital funcional e inovadora alinhada à metodologia **Sinfonia**.
  2. Avaliar a aplicação prática da metodologia, incluindo propostas de melhorias e adaptações.
  3. Utilizar controle de versão (Git) para registrar a evolução do projeto e a colaboração entre membros.
  4. Integrar princípios de engenharia de software, práticas éticas e estratégias de implantação em produção.
- 

## 3. Diretrizes e Regras

1. **Formação de Equipes:** Cada equipe deve ter entre 3 e 6 membros.
2. **Metodologia:** A metodologia **Sinfonia** é obrigatória e deve ser documentada como parte do projeto.
3. **Incrementalidade:** O sistema deve ser desenvolvido de forma incremental, respeitando os marcos e checkpoints previstos.

4. **Líder de Equipe:** Cada equipe deve designar um líder responsável pela comunicação com o professor.
  5. **Gestão de Projetos:** Utilize ferramentas de organização, como GitHub Projects, Trello ou Jira, para planejar e acompanhar tarefas.
  6. **Feedback:** Feedbacks serão fornecidos periodicamente e deverão ser incorporados no projeto.
  7. **Experimentação:** As equipes podem criar novos artefatos e propor alterações na metodologia, documentando justificativas e lições aprendidas.
- 

## 4. Ferramentas e Recursos Recomendados

- **Versionamento:** GitHub ou GitLab.
  - **Desenvolvimento:** Frameworks e linguagens de escolha da equipe (ex.: Python, Node.js, PHP).
  - **Documentação:** Ferramentas como Lucidchart ou Mermaid para diagramas, Google Docs para edição colaborativa e Markdown para documentação técnica.
  - **Entrega e Implantação:** Serviços como Vercel, Netlify, Heroku, AWS, Azure ou similares para hospedar a solução final. Ferramentas como Docker e Kubernetes para empacotamento (opcional).
- 

## 5. Requisitos do Projeto

1. **Imersão:**
  - Seleção do problema de negócio utilizando o **Canvas de Identificação do Domínio**.
  - Detalhamento das metas e indicadores no **Canvas de Objetivos de Projeto**.
2. **Ideação:**
  - Desenvolvimento de soluções no **Canvas de Ideação de Soluções**.
  - Estruturação de prompts com o **Canvas de Design de Prompts**, se aplicável.
3. **Produção:**
  - Documentação arquitetural utilizando o **C4 Model** (nível Contexto, Contêiner e Componente; nível Código é opcional).

- Implementação da solução com versionamento Git, incluindo histórico de commits demonstrando colaboração.
- Testes e validações registrados no **Canvas de Testes e Validação**.
- Implantação da solução em ambiente de produção, com um host funcional.

#### 4. **Validação:**

- Análise de escalabilidade e diversificação com os respectivos canvases.
- Reflexão sobre ética e impacto social documentada no **Canvas de Feedback e Iteração**.

---

## 6. Critérios de Aceitação

1. **Relevância:** A solução atende ao problema identificado e agrega valor ao negócio.
2. **Qualidade Técnica:** Arquitetura robusta, modelo treinado e validado, código-fonte versionado e documentado.
3. **Documentação Completa:**
  - Organização e clareza do repositório.
  - Artefatos da metodologia preenchidos e alinhados ao projeto.
  - Modelagem e prototipação (wireframes, mockups ou protótipos Hi-Fi) - se pertinente.
  - Documentação técnica abrangendo decisões, design, e regras de negócio.
  - Código-fonte versionado, bem estruturado e documentado.
  - Casos de teste e validação, versionados junto ao código.
4. **Produção:**
  - Progresso incremental, com entregas intermediárias respeitando os marcos do cronograma.
  - Engajamento da equipe e qualidade da comunicação (evidências em reuniões e feedbacks).
  - Sistema funcional implantado em um host acessível e demonstrável.
5. **Engajamento Metodológico:** Registro de lições aprendidas, reflexões sobre a metodologia e propostas de melhorias.

---

## 7. Organização do Repositório

O repositório do projeto deve ser estruturado como um **portal centralizado para visitantes**, inspirado nas boas práticas do artigo [Let me in: Guidelines for the](#)

[Successful Onboarding of Newcomers to Open Source Projects](#) (STEINMACHER, et al., 2018). O repositório deve conter:

- **README.md:** Página inicial clara e bem estruturada contendo:
  - Descrição geral do projeto.
  - Objetivos principais e funcionalidades esperadas.
  - Estrutura organizacional do código, com diagramas (ex.: C4).
  - Links para recursos importantes (ex.: workspace, ferramentas de revisão de código, rastreador de problemas).
  - Guia para build local do sistema.
  - Lista de issues, com marcação para tarefas iniciais (ex.: labels como *good first issue*).
  - Orientações sobre como contribuir para o projeto.
- **CONTRIBUTING.md:** Guia detalhado para novos colaboradores:
  - Como configurar o ambiente de desenvolvimento.
  - Processo de submissão de código e revisão.
- **BUILD.md:** Instruções claras para construir e executar o sistema localmente.
- **Diretório de Diagramas:** Diagramas (ER, C4, entre outros) para facilitar o entendimento da estrutura do sistema.

Esses elementos serão avaliados como parte da organização e comunicação do projeto.

---

## 8. Cronograma

1. **Definição do Problema:** Aula 7.
  2. **Checkpoint 1:** Aula 12 – Primeira apresentação e feedback - Imersão e Ideação.
  3. **Checkpoint 2:** Aula 20 – Revisão intermediária e ajustes.
  4. **Entrega Final:** Aula 26 – Apresentação do projeto.
  5. **Avaliação e Reflexão:** Aulas 27 e 28 – Feedback detalhado e lições aprendidas.
-