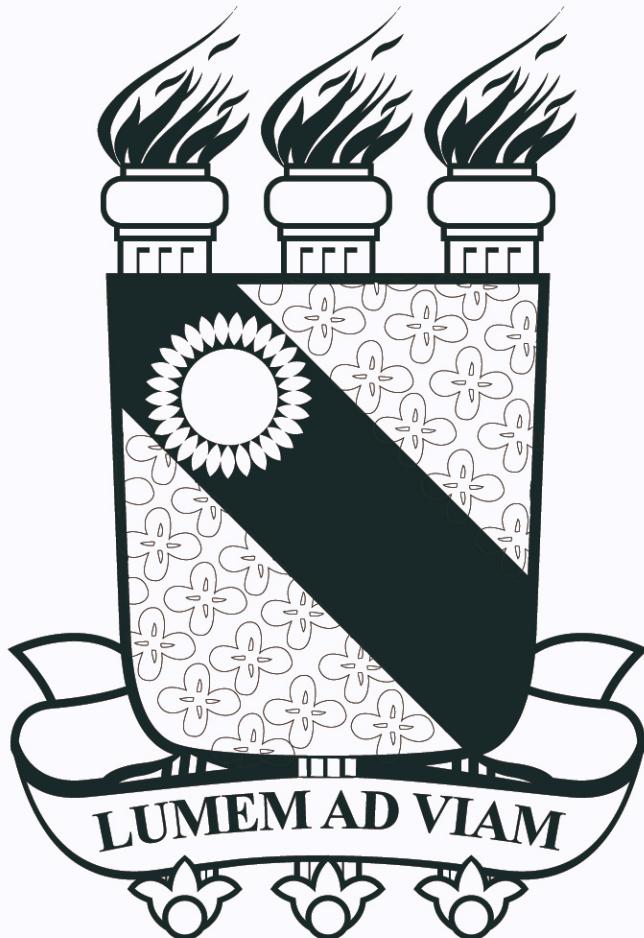


Relatórios do Projeto Research Flow



**UNIVERSIDADE
ESTADUAL DO CEARÁ**

Equipe:

1. Bianca Leão
2. Glauco Cipriano Moreira
3. João Victor dos Santos
4. Pedro Mikhael Maia de Souza
5. Rian Vilanova de Oliveira Souza

Dificuldades Encontradas e Lições Aprendidas **(1ª apresentação)**

Este tópico visa analisar as principais dificuldades enfrentadas durante o desenvolvimento do projeto Research Flow na 1ª apresentação, detalhando as estratégias de mitigação adotadas e as lições aprendidas em cada área crítica.

1. Desenvolvimento do front end(Uso do React)

Desafio:

A equipe de Front-End enfrentou dois principais desafios. O primeiro foi lidar com as constantes mudanças no visual e nas funcionalidades do projeto, para que todos do grupo ficassem satisfeitos com o resultado. Isso gerou vários ajustes e retrabalhos ao longo do processo.

O segundo desafio foi o uso do React. A biblioteca era nova para todos da equipe de Front-End, o que exigiu tempo para aprender como criar componentes, organizar o código e aplicar os conceitos na prática.

Estratégia e Resultado:

Para organizar melhor as mudanças e garantir que todos estivessem alinhados, a equipe realizou reuniões frequentes para receber feedback do grupo. Isso ajudou a corrigir problemas mais rápido e deixou o desenvolvimento mais claro para todos.

No aprendizado do React, a equipe contou com o apoio de ferramentas de IA, como ChatGPT e Gemini, que ajudaram a tirar dúvidas e entender melhor como aplicar a biblioteca no projeto.

Com isso, o Front-End evoluiu de forma mais segura, ficando mais organizado, funcional e pronto para a apresentação.

2. Coordenação e Dinâmica de Equipe

Desafio:

A quantidade da equipe e a inexperiência em trabalhar em um projeto com vários colaboradores dificultou a organização, causou atrasos nas reuniões e deixou o ritmo do desenvolvimento mais lento no início.

Estratégia e Resultado:

Para resolver isso, a equipe dividiu as responsabilidades por áreas, como Backend, Front-End e outras partes específicas do projeto. Também passaram a usar ferramentas de organização e combinaram uma forma clara de comunicação entre todos. Além disso, as reuniões passaram a ser feitas apenas para resolver problemas e revisar o que já tinha sido feito.

Com essas mudanças, o trabalho ficou mais organizado, o grupo se comunicou melhor e o desenvolvimento avançou de forma mais eficiente.

Dificuldades Encontradas e Lições Aprendidas (Entre a 1^a e 2^a Apresentação)

Este tópico visa analisar as principais dificuldades enfrentadas durante o desenvolvimento do projeto Research Flow no intervalo entre a primeira e a segunda apresentação, detalhando as estratégias de mitigação adotadas e as lições aprendidas em cada área crítica.

1. Otimização do Desempenho da LLM e Gestão de Tokens

Desafio:

A seleção e otimização dos modelos de Large Language Model (LLM) da família Gemini representaram um desafio central. Especificamente, a necessidade de balancear a precisão (qualidade da revisão e resumo de artigos) com a velocidade de resposta e a eficiência na gestão de recursos, sobretudo o limite de tokens, foi crítica. A limitação de tokens manifestou-se rapidamente em testes com documentos PDF de maior volume.

Estratégia e Resultado:

Foram realizados testes comparativos exaustivos com diferentes versões do Gemini. O modelo Gemini 2.5 Pro demonstrou ser o mais adequado para tarefas de maior complexidade cognitiva, como a revisão aprofundada e a criação de resumos concisos e academicamente rigorosos. Para a funcionalidade de pesquisa e rastreamento inicial de artigos, o Gemini 2.5 Flash foi adotado devido à sua alta velocidade e custo-benefício na utilização de tokens. Esta abordagem dual permitiu otimizar a alocação de recursos computacionais por funcionalidade.

Modelo Gemini	Aplicação Principal	Características Chave
2.5 Pro	Revisão e Criação de Resumos.	Alta precisão e rigor acadêmico.
2.5 Flash	Pesquisa e Rastreamento de Artigos.	Alta velocidade e eficiência de tokens

2. Coordenação e Dinâmica de Equipe

Desafio:

A dimensão da equipe e a inexperiência prévia dos membros em projetos de extensão com múltiplos colaboradores introduziram complexidades na coordenação. Isso impactou a frequência e a eficácia das reuniões de alinhamento e o ritmo do desenvolvimento das entregas.

Estratégia e Resultado:

A estruturação de responsabilidades em módulos específicos (e.g., Backend, Frontend, Otimização de LLM) e a implementação de ferramentas de gestão de projetos foram cruciais. Foi estabelecido um protocolo de comunicação assíncrona robusto, complementado por reuniões síncronas focadas apenas em bloqueios e revisões de sprint, o que melhorou significativamente o fluxo de trabalho.

3. Integração de Software Específico para Manipulação de Documentos

Desafio:

A necessidade de garantir a correta compilação e formatação do output gerado pelo Gemini, que se dava em formato .tex (LaTeX), para um PDF visualmente apresentável e compatível com o FrontEnd exigiu a integração de um novo software. O MilkTek foi a solução escolhida, demandando uma curva de aprendizado inicial e um esforço considerável para configurar a rota de formatação de PDFs de forma estável e precisa.

Estratégia e Resultado:

A equipe dedicou horas ao aperfeiçoamento da pipeline de processamento que envolve a recepção do arquivo .tex do Gemini, sua manipulação via MilkTek, e a entrega final ao FrontEnd. Essa fase resultou em um módulo de formatação robusto, capaz de gerar documentos padronizados e academicamente aceitáveis a partir da saída da LLM.

Dificuldades Encontradas e Lições Aprendidas **(Entre a 2^a e a 3^a apresentação)**

Este relatório busca expor as dificuldades e mudanças que tivemos que lidar desde a apresentação anterior até agora. Essas mudanças ao longo do projeto nos auxiliaram a adquirir diversos conhecimentos no desenvolvimento de uma ferramenta front-end com uso da API do gemini para desenvolvimento das funcionalidades do projeto.

1. Mitigar a geração de tags em .tex pelo Gemini incompatíveis com a compilação para PDF:

Desafio: de alguns dias até a entrega final do projeto, nós vimos que o modelo gemini-2.5-pro da API deixou de ser gratuito e por isso passamos a usar apenas o gemini-2.5-flash em todas as funcionalidades. O problema é que por ele ser mais simples, muitas vezes ele alucinava muito e gerava códigos para .tex incompletos ou completamente desconexos.

Por exemplo: um caso se repetiu várias vezes, era que ele abria um `\begin{itemize}` aninhado, sempre abrindo um dentro do outro e no final não fechava todos corretamente com `\end{document}`, isso gera erro de compilação e ele não cria o PDF.

Resolução: para resolver esse problema, tive que enriquecer o prompt da funcionalidade para que proibisse o modelo de aninhar muitas dessas tags para diminuir a chance de dar erro no fechamento, além de deixar claro que ele sempre tem que fechar elas também.

2. Troca dos modelos gemini-2.5-pro em algumas funcionalidades:

Desafio: Além do Formatar Artigo, tive que melhorar o prompt do Resumo Estruturado também, pois agora sem o Gemini Pro, o retorno recebido pela API muitas vezes era “menos inteligente/completo” já passamos a usar a versão flash. Assim tive que deixar muitas coisas mais claras no prompt para que o modelo não continuasse a cometer erros.

Prompt Formatador:

```
prompt = f"""
You are a LaTeX formatting specialist. Your task is to convert the text below into LaTeX.

STRICT RULES (To avoid breaking the compiler):
1. The code will be injected into a file that ALREADY HAS \\documentclass and \\begin{document}.
2. IMAGES ARE FORBIDDEN: Do not use '\\includegraphics' or 'figure'. If there is any mention of an image, write only "[Image removed]".
3. COMPLEX TABLES ARE FORBIDDEN: Do not use the 'tabular' or 'table' environment. Convert all tables into lists ('itemize' or 'enumerate') or descriptive text.
4. UNICODE MATH IS FORBIDDEN: Do not use symbols like  $\theta$ ,  $\beta$ ,  $\infty$ . You MUST use the commands '$\\alpha$', '$\\beta$', '$\\infty$'.
5. DO NOT include any preamble (no \\documentclass, no \\usepackage).
6. Do NOT include \\begin{document} or \\end{document}.
7. Start directly with \\section{Title} or with the content.
8. For mathematics, use ONLY LaTeX commands ($\\alpha$, $\\beta$), NOT Unicode symbols.
9. TEXT ALIGNMENT: The body text MUST be fully justified (standard LaTeX behavior).
10. PROHIBITED: Do NOT use the commands '\\centering' or '\\begin{center}' for the main body text. Only center the Main Title if necessary, then immediately switch back.
11. NO DEEP NESTING: Do NOT nest lists ('itemize' or 'enumerate') more than 2 levels deep. Deep nesting causes the compiler to crash ("Too deeply nested" error).
12. CLOSE ALL ENVIRONMENTS: Every '\\begin{...}' MUST have a matching '\\end{...}'. Double-check that no list is left open.
13. ESCAPE SPECIAL CHARACTERS: You MUST escape reserved characters in text mode to avoid "Runaway argument":
    - '%' becomes '\\%'
    - '_' becomes '\\_'
    - '&' becomes '\\&'
    - '$' becomes '\\$'
    - '#' becomes '\\#'
14. Desired style: {few_shot}.

Original Text:
{input_text[:50000]}

The API must translate the final response into Portuguese.
....
```

Prompt Resumo:

```
# 1. Instrução de foco
instruction_block = ""
if natural_language_query and natural_language_query.strip():
    instruction_block = f"""
        INSTRUÇÃO PRIORITÁRIA DO USUÁRIO: "{natural_language_query}".

        Você deve filtrar e adaptar os campos abaixo para focar nesta consulta.
        Se o artigo não responder à consulta, explice isso na conclusão.
    """
else:
    instruction_block = "Gere um resumo técnico e abrangente."

# 2. Prompt principal
prompt = f"""
    Você é um especialista em síntese científica. Analise o texto fornecido e preencha os campos solicitados.

{instruction_block}

Preencha os campos em Português do Brasil de forma detalhada.
Texto do Artigo:
{article_text[:60000]}
"""
```

3. Adição da consulta do usuário do Resumo:

Problema/Resolução: Anteriormente não tinha como receber uma consulta do usuário apesar de existir a possibilidade do back-end receber. Isso pois o frontend não possuía uma área dedicada para receber essa “query”. Agora essa área para consulta existe e é sim passada para o modelo.

4. Criação do Banco de Dados:

Desafio: Configuração correta do ambiente e das migrações do banco de dados, o que gerou dúvidas iniciais se as tabelas e relacionamentos estavam sendo criados e persistidos corretamente.

Resolução: Desenvolvimento de scripts de verificação direta no banco para validar os dados sem depender da interface, além da padronização de caminhos e uso rigoroso de variáveis de ambiente.