

Planejamento e Gestão de Riscos - RedaPlus

Planejamento

Para o desenvolvimento do projeto, foram elicitados 6 grandes requisitos, sendo eles:

1. API;
2. Frontend para consumir a API;
3. Integração com LLM;
4. Testes Unitários;
5. Testes de Integração;
6. Pipeline.

Todos esses pontos precisavam ser implementados em **até cinco dias corridos**. Com os requisitos e o prazo bem estabelecidos, a escolha das tecnologias e bibliotecas foi o próximo passo:

- **API:** Optou-se pelo uso do NestJS (Node.js) devido à sua arquitetura modular, que organiza o código em controllers e services, promovendo escalabilidade e mantendo o projeto limpo e bem estruturado.
- **Frontend:** Considerando o tempo reduzido, foi utilizado Next.js (React) em conjunto com a biblioteca MUI, que oferece diversos componentes pré-estilizados. Essa escolha aumentou significativamente a produtividade e ajudou a cumprir o cronograma.
- **Integração com LLM:** Para automatizar e abstrair a comunicação entre a LLM, a API e o Frontend, foi adotada a ferramenta **N8N**, facilitando a orquestração entre os serviços.
- **Testes:** Os testes unitários foram implementados com Jest, enquanto os testes de integração foram realizados utilizando o Cypress.

Foram desenvolvidos pipelines utilizando o GitHub Actions para realizar o build e a execução dos testes unitários do back-end, bem como o build do front-end.

Alocação de Recursos

- **Backend Developer:** 2 dias (API + testes unitários)
- **Frontend Developer:** 1,5 dia (Next.js + MUI + testes + integração)
- **DevOps/Integração:** 1 dia (Pipelines + N8N + automação com LLM)
- **Gerência de Projeto/Testes:** 0,5 dia (planejamento + trello + revisão)

Análise de Risco

Antes do Desenvolvimento, foi pensado possíveis problemas que poderiam acontecer no cenário desse teste e uma estratégia de mitigação inicial foi proposta:

Risco	Impacto	Estratégia de Mitigação
Extrapolar o limite de tokens com a LLM	Alto	Padronizar prompt + limitar entrada (1000-2000 tks)
Integração da LLM falhar	Médio	Isolar via N8N para facilitar troca/substituição

Interdependências

- O **frontend** depende da **API** estar disponível e estruturada.
- A **LLM** depende do funcionamento conjunto entre **API**, **Frontend** e **N8N**.
- O sucesso dos **testes de integração** depende da estabilidade da API e da consistência do frontend.
- Os **pipelines** automatizados são essenciais para garantir que alterações em qualquer parte do sistema não quebrem o fluxo de uma futura entrega contínua.

Para controle das atividades, foi utilizado o [trello](#)