

# EXAMFORGE

Entrega Final



EXAMFORGE

# EQUIPE

**Lyedson**Requisitos, Front-end e  
Geral**Larissa Kelly**Requisitos, Front-end e  
Testes**Marina**

Design e Front-end

**José Fortunato**

Back-end e IA

**Gabriel Marques**

Back-end e IA



EXAMFORGE

# INTRODUÇÃO

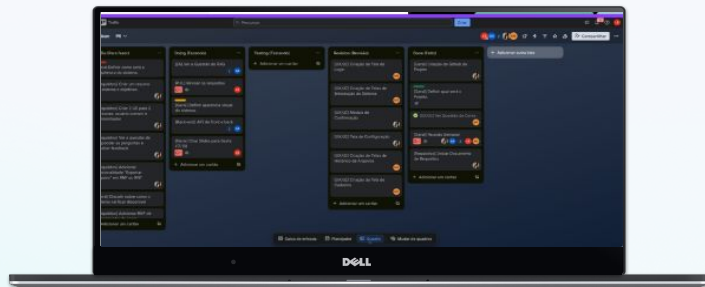
“O EXAMFORGE é uma ferramenta web desenvolvida para ser um recurso de produtividade e apoio ao estudo, com foco na geração inteligente de questões a partir de documentos. ”



**Objetivo:** Automatizar a criação de avaliações acadêmicas, transformando materiais de estudo em simulados personalizados de forma rápida e inteligente.

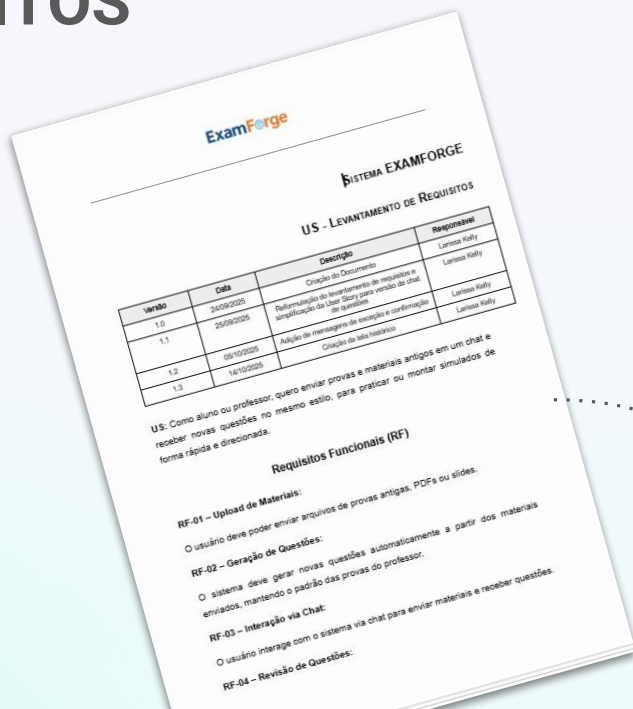
EXAMFORGE

# PLANEJAMENTO

**Ferramenta Organizacional** **Trello****Ferramenta para Encontros Síncronos** **Discord**

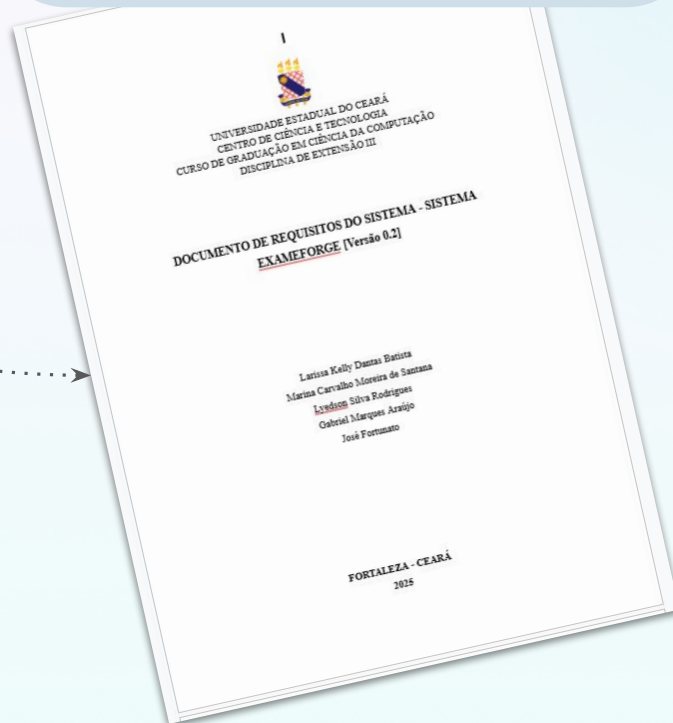
EXAMFORGE

# REQUISITOS



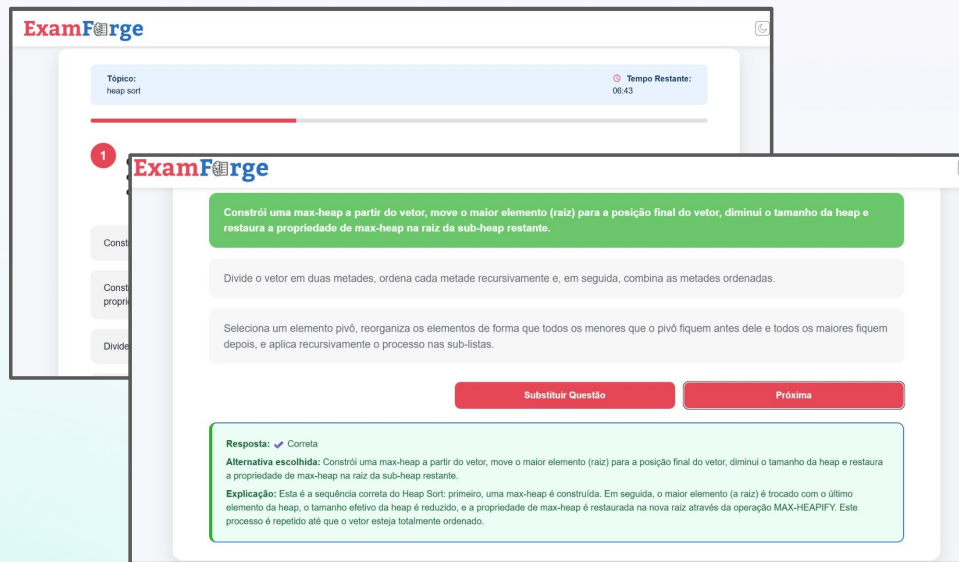
**Requisitos v0.1**  
Mais simples, somente RF, RNF

**Requisitos v0.2**  
Focamos em RN, RNF e RN e Arquitetura do Sistema



## EXAMFORGE

# FRONT-END



A versão final do projeto conta com uma série de melhorias em relação à versão anterior:

- Paleta de cores mantida com ajustes sutis
- Cor de destaque das questões alterada de vermelho para verde (evita confusão visual)
- Estados de loading adicionados durante:
  - Geração das questões
  - Vetorização dos documentos
- Feedback visual durante processos demorados (antes não havia indicação de andamento)

## EXAMFORGE

# BACK-END

A versão final do projeto conta com uma série de melhorias em relação à versão anterior:

- Qualidade das questões significativamente melhorada;
- Adicionado funcionalidade de gerar PDF das questões;
- Avaliação detalhada do desempenho do usuário ao final do simulado;
- Adicionado função de substituir uma questão gerada
- Base de dados alterada de forma a evitar vetorização duplicação de documentos

## Questão 1:

Qual é a principal vantagem da arquitetura [Long Short-Term Memory \(LSTM\)](#) em comparação com redes neurais recorrentes (RNNs) tradicionais sequenciais?

- LSTMs utilizam um número significativo de parâmetros, o que reduz o risco de overfitting.
- A capacidade das LSTMs de regular o fluxo de informação através de forget e output gates permite reter e descartar informações de longo prazo, evitando o problema do gradiente evanescente e capturando dependências de longo prazo.
- LSTMs eliminam completamente a necessidade de inicialização cuidadosa, permitindo uma forma independente para maior paralelização.
- LSTMs são intrinsecamente mais rápidas devido à sua arquitetura simplificada.

## Questão 2:

Em uma rede LSTM, qual é a função primária do esquecimento e saída interagindo com ela?

- A célula de memória armazena apenas a saída final, sem influenciar o fluxo de informação.
- A célula de memória retem e atualiza informações enquanto as portas de entrada, esquecimento e saída controlam o fluxo de informações para a célula.
- A célula de memória é responsável por adicionar ruído ao sinal de entrada.

## Questão 1:

Qual das seguintes afirmações descreve corretamente a complexidade de tempo do algoritmo Quicksort?

- É sempre linear ( $O(n)$ ) em todos os casos, tornando-o consistentemente o mais rápido.
- Apresenta complexidade quadrática ( $O(n^2)$ ) no pior caso, mas é linear em média.
- Sua complexidade é sempre logarítmica ( $O(\log n)$ ), independentemente da entrada.
- É um algoritmo elementar, com desempenho similar a métodos de ordenação simples em todas as situações.

## Questão 2:

No algoritmo Heap Sort, a função MAX-HEAPIFY é fundamental para manter a propriedade de heap. Qual é a ação principal realizada por MAX-HEAPIFY quando um nó é menor que seu filho, violando a propriedade de max-heap?

- Inserir o nó em uma nova posição na raiz do heap, reconstruindo a estrutura.
- Remover o nó e todos os seus descendentes, pois a estrutura do heap foi comprometida.
- Trocar o nó com o filho maior e mover-se para baixo na árvore, continuando até que o nó não seja menor que seus filhos.
- Trocar o nó com o filho menor e mover-se para cima na árvore até que a propriedade seja restaurada.

## Questão 3:

O algoritmo Quicksort, inventado por C.A.R. Hoare, baseia-se em um problema fundamental para realizar a ordenação de um vetor. Qual é esse problema central?

- A fusão de subvetores já ordenados, combinando-os em um único vetor maior.



## EXAMFORGE

# IA



Seguimos com o Gemini 2.5 Flash do Google. Os avanços na geração das questões vieram, principalmente, de melhorias no prompt do sistema:

- Estabelecemos limites mais claros para evitar que a IA fugisse dos conteúdos dos documentos;
- Demos mais espaço para sua interpretação, deixando-a usar sua inteligência de forma mais natural;
- Permitimos maior complexidade nas questões, que agora avaliam raciocínio, não só memorização.

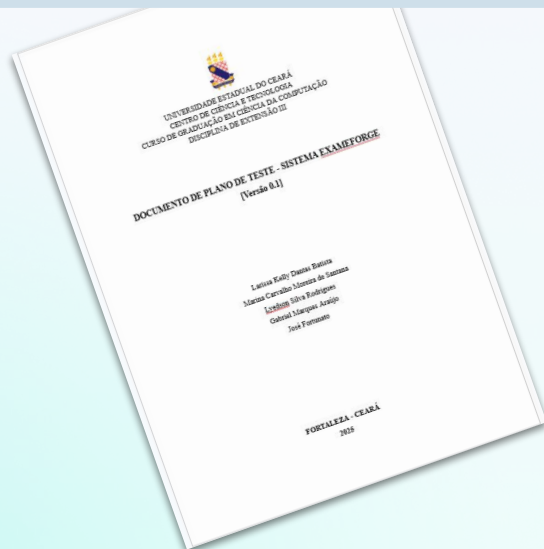
A IA se mostrou consistente na maioria dos testes realizados. As poucas alucinações que ela teve foram remediadas a partir de pequenas alterações no prompt



EXAMFORGE

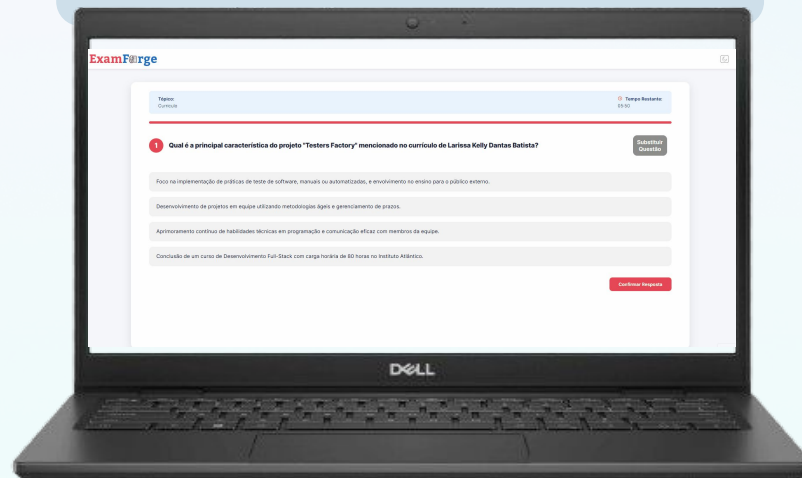
# TESTES

**Documentos de Plano de Teste**  
Casos, cenários de teste



**Testes Exploratórios**  
Testes exploratórios na ferramenta

**Reporte e Reteste de Bugs**  
Reportados Bugs e Retestados

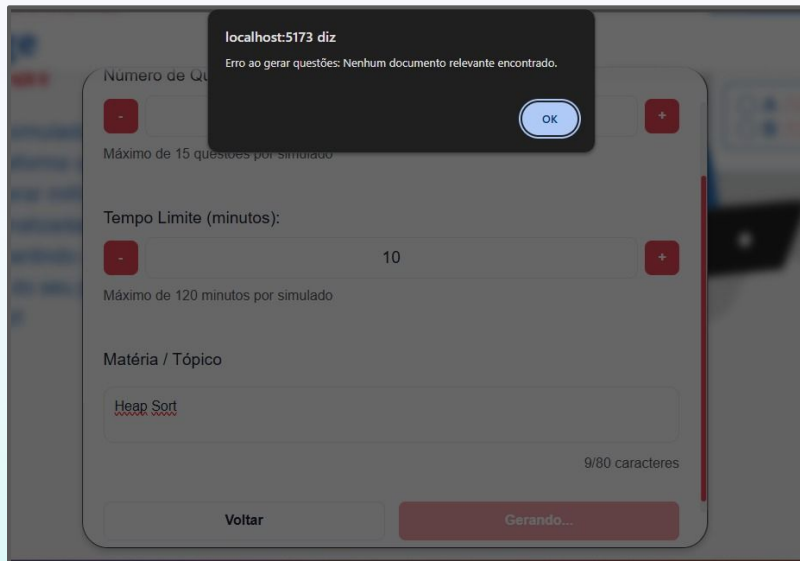


## EXAMFORGE

# DIFICULDADES

Nossas principais dificuldades na entrega atual foram:

- Limitação de tokens ao testar arquivos grandes, arquivos extensos não puderam ser processados integralmente;
- Problemas de compatibilidade do backend, só funcionou corretamente para os testes após a instalação do Python 3.11;
- Backend inicial criado sem ambiente virtual (venv);
- Instalação demorada das dependências.



**PERGUNTAS?**

**OBRIGADO!**