UNIVERSIDADE ESTADUAL DO CEARÁ
CENTRO DE CIÊNCIA E TECNOLOGIA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA DE EXTENSÃO III

RELATÓRIO - SISTEMA EXAMEFORGE

Gabriel Marques Araújo
José Fortunato Mendonça
Larissa Kelly Dantas Batista
Lyedson Silva Rodrigues
Marina Carvalho Moreira de Santana

FORTALEZA - CEARÁ
2025

Universidade Estadual do Ceará

ExamForge

Conteúdo

1. Introdução

Este relatório apresenta a trajetória de desenvolvimento do projeto ExamForge, detalhando desde a definição inicial da proposta até as versões mais recentes da aplicação. Ao longo do documento, são descritas as decisões técnicas, os desafios enfrentados, as soluções adotadas e as evoluções obtidas nas diferentes fases do projeto. O objetivo é registrar, de forma organizada e objetiva, todo o processo de construção da plataforma, desde os estudos iniciais sobre LLMs e RAG, passando pelo desenvolvimento do front-end e back-end, até as melhorias implementadas e os próximos passos previstos.

2. PARTE I - 17/10/2025

2. 1 Definição do Projeto

No início do projeto, nós da equipe pensamos em desenvolver uma plataforma web, na qual o usuário pudesse realizar seu cadastro e gerar simulados com base em materiais de estudo que ele fornecesse separados por áreas do conhecimento. A ideia era que a LLM interpretasse o conteúdo e desenvolvesse simulados personalizados, contendo questões objetivas e subjetivas, de acordo com o perfil e interesse do usuário, permitindo que o usuário salvasse e organizasse seus simulados dentro da plataforma por disciplinas, podendo acessar, editar e refazer esses simulados posteriormente.

Essa proposta inicial visava oferecer uma experiência ampla e flexível, integrando diversas funcionalidades relacionadas ao estudo autônomo. Entretanto, durante as reuniões internas e os primeiros estudos técnicos, a equipe identificou diversas dificuldades e limitações importantes para o desenvolvimento desta proposta, tais como:

- Escopo muito amplo, inviável para o tempo de desenvolvimento disponível na disciplina e para a capacidade de entrega do grupo.
- Concorrência com ferramentas já consolidadas, como o *ChatGPT*, que já oferecem geração de questões e textos de forma mais robusta.
- Desafio técnico de criar um modelo próprio capaz de gerar diferentes quantidades e tipos de questões sem alucinações, além de integrá-las ao front-end de maneira dinâmica, uma vez que não haveria padronização nas telas.
- Ausência de um público-alvo bem definido, já que não estava claro se a aplicação seria voltada para alunos, professores ou o público em geral.

Link do slide da primeira proposta: Canva

Diante dessas dificuldades, após novas discussões e alinhamentos, nós da equipe decidimos repensar o escopo do projeto com o objetivo de torná-lo mais objetivo, funcional e condizente com o tempo disponível. Assim, chegamos à proposta atual: Uma plataforma web que não necessita de cadastro e tem como foco principal a geração de simulados de múltipla escolha baseados em materiais fornecidos pelo usuário, permitindo o usuário exportar como pdf para estudo.

Nessa nova proposta, o usuário pode:

- Fazer o upload de seus materiais de estudo;

- Definir a quantidade de questões, o tempo de realização e se haverá múltiplas alternativas corretas;
- Realizar o simulado diretamente na plataforma;
- Receber o resultado com correção automática e explicações detalhadas das respostas;
- E, ao final, exportar o simulado em PDF para uso posterior ou anotações.

Essa reformulação tornou o projeto mais enxuto, viável e direcionado, mantendo a essência da proposta inicial, que seria o uso de IA como apoio ao aprendizado, mas com uma implementação mais realista e funcional dentro do escopo da disciplina.

Link do slide da reformulação da proposta: Slide Reformulado

Link Protótipo: Figma

2.2 LLMs, RAG e Engenharia de Prompt

Como a maior parte dos membros da equipe ainda não possuía familiaridade com a implementação de modelos de linguagem (LLMs) ou aplicações que fizessem a sua utilização, as primeiras semanas do projeto foram dedicadas a estudos teóricos e práticos sobre o funcionamento dessas ferramentas, incluindo RAG (Retrieval-Augmented Generation) e técnicas de fine-tuning e engenharia de prompt.

Os primeiros testes foram realizados com um modelo sem *system prompt* bem definido. Solicitamos que o modelo gerasse cinco questões de múltipla escolha sobre um tema definido. O resultado foi o esperado: o modelo alucinou, retornando respostas fora do escopo solicitado e, em alguns casos, respostas em inglês, demonstrando a necessidade de maior controle no contexto da geração

Nos testes seguintes passamos a realizar com prompts mais elaborados, mas ainda sem o uso de RAG.O *prompt* já incluía uma persona para o modelo, que foi definida como: “Assistente virtual com foco em criação de questões de múltipla escolha”, seguido de um exemplo de estrutura de questões, servindo como guia de formatação. Os resultados foram mais satisfatórios, já era seguido o padrão estabelecido. O modelo passou a retornar o número correto de questões, todas estruturadas e coerentes com o padrão estabelecido.

Em seguida, o mesmo modelo foi testado com o uso de RAG, permitindo requisições mais específicas a partir de arquivos reais (PDFs de disciplinas). Os retornos iniciais foram satisfatórios, com as questões sendo geradas de forma consistente e sem alucinações. Em síntese, essas foram as decisões mais importantes que tivemos nessa fase:

- Os *system prompts* devem ser bem definidos e objetivos, contendo exemplos de estrutura esperada das questões.
- O uso de RAG será adotado como estratégia principal para gerar questões baseadas no conteúdo enviado, por atender de forma eficaz às necessidades do projeto.

Ainda há alguns pontos a serem discutidos, como a escolha definitiva do modelo de linguagem mais adequado para o nosso objetivo, qual base de dados é a melhor opção e se existe alguma maneira de melhorar o prompt, mas a base principal do projeto já está definida.

2.3 Próximos Passos

1. Dar continuidade ao desenvolvimento do projeto, a partir do protótipo apresentado em sala, verificando a viabilidade de implementar as funcionalidades de:
    a. Geração automática de questões de múltipla escolha;
    b. Controle de tempo (temporizador) durante os simulados;
    c. Envio e validação de arquivos pelo usuário;
    d. Exportação do simulado em formato PDF.
2. Estudar a viabilidade de novas funcionalidades, como a inclusão de questões subjetivas e variações no formato das questões de múltipla escolha.
3. Explorar o uso da FastAPI, visando facilitar a integração entre o frontend e os serviços de geração de questões via LLM.
4. Analisar e propor melhorias no sistema de correção, tornando-o mais preciso e ajustado às respostas fornecidas pelo usuário.
5. Iniciar o desenvolvimento do frontend, definindo a identidade visual e o design da interface, com base no protótipo construído no Figma.
6. Finalizar a documentação inicial do projeto, consolidando os requisitos e elaborando o diagrama de sequência representando o fluxo geral do sistema.

3. PARTE II - 14/11/2025

3.1 Desenvolvimento do projeto

Com os requisitos principais do projeto já estabelecidos, iniciamos o desenvolvimento da próxima versão. O protótipo mostrado na primeira apresentação era um *chatbot* que lia um documento enviado e gerava uma questão única. Após a resposta, o modelo também retornava uma pequena resolução.

A segunda versão é uma aplicação web que gera múltiplas questões. O usuário tem a opção de enviar um ou mais documentos, que serão chunkenizados, vetorizados e armazenados na base de dados. Cada documento tem um número específico de chunks, que poderão ou não ser retornados dependendo do conteúdo que o usuário inserir no campo “tópicos”, campo esse que será vetorizado e utilizado no processo de Retriever, que irá recuperar o maior número de contéudo relacionado ao tópico (RAG). O modelo receberá o que for retornado pelo RAG e, juntamente com o prompt de sistema, gerará uma resposta no formato de JSON contendo as questões e a resolução. O usuário também possui as opções de definir o número de questões a serem geradas para o seu simulado e o tempo para a realização do simulado.

Para o front, com o figma já definido demos inicio a produção das telas. Nesse ponto ainda não estava totalmente definido exatamente qual paletas de cores seria utilizada, no momento estamos discutindo internamente o que será feito.

Link dos slides: Slides Apresentação II

3.2 Decisões importantes

Algumas decisões importantes que envolveram essa entrega foram:

1. **Back-end:** Para o desenvolvimento do Back-end, optamos pelo uso do FastAPI (framework Python), pela facilidade de uso e pela facilidade de integração de ferramentas de IA com

python. Nem todos os membros do grupo estavam familiarizados com essa ferramenta, então cedemos alguns dias apenas estudando aplicações com ela.

2. **Front:** Para o Front usamos principalmente TypeScript, com foco em bibliotecas como Axios e React. Optamos pela utilização dessas ferramentas pois existem integrantes do grupo que possuem familiaridade com ela, o que economizou certo tempo de desenvolvimento.
3. **Modelo IA e embeddings:** O modelo de Inteligência artificial escolhido foi o gemini-2.5-flash da Google. O modelo foi escolhido exclusivamente por conveniência, já que o gemini pro foi oferecido de graça para estudantes a algum tempo atrás, o que nos permite utilizar mais tokens. Os embeddings usados também foram do Google, sendo o gemini-embedding-001 o escolhido para nossa aplicação.
4. **Gerenciamento de projeto:** Para garantir a organização e o acompanhamento contínuo do projeto, estabelecemos que realizaremos reuniões semanais às terças-feiras, utilizando a plataforma Discord. Quanto à gestão de tarefas, optamos pela metodologia Kanban, devido à sua flexibilidade e adaptabilidade às dinâmicas do nosso fluxo de trabalho

3.3 Dificuldades

As principais dificuldades nessa entrega foram:

1. **Integração do código:** Não tínhamos muita familiaridade com o desenvolvimento de projetos, então no início foi bem desafiador integrar o front com o back. Para evitar conflitos iniciais, optamos por criar duas branches diferentes no repositório virtual, sendo uma para o back e outra para o front, o que nós permitiu avançar o desenvolvimento de forma bem consistente. A integração foi feita após terminados o desenvolvimento das features de geração de questão e resolução, no back, e após a finalização das telas iniciais, no front. Após um pouco de dor de cabeça, a aplicação funcionou muito bem.
2. **Criação das telas:** As telas passaram por várias mudanças ao decorrer do desenvolvimento e o motivo disso foi principalmente a falta de comunicação. Eram feitas alterações que não agregavam em nada e alteravam a proposta inicial do figma. Esse problema foi resolvido após algumas reuniões e conversas privadas, que alinharam os objetivos dos membros.

3.4 Próximos passos

Os próximos passos serão:

1. Alterar pequenos detalhes do front, como as cores da questão selecionada (na versão atual, quando a alternativa é selecionada ela fica destacada em vermelho, o que pode confundir o usuário).
2. Implementar features de:
    a. 2.1 Substituir uma questão gerada (sugerido por um dos examinadores).
    b. 2.2 Gerar PDF com as questões.
    c. 2.3 Avaliação final do simulado, contendo pontos que devem ser reforçados
3. Continuar a documentação da arquitetura do projeto.
4. Otimizar o processo de vetorização e criação da base de dados

4. PARTE III - 05/12/2025

4.1 O que foi feito

Demos continuidade do que foi especificado no último documento. Muitas features importantes que não estavam na versão anterior do projeto foram implementadas. Entre elas:

- Geração do PDF do simulado
- Opção de substituição de questão
- Avaliação final detalhada com os pontos que devem ser priorizados

Além disso, outras features já existentes na versão anterior foram aprimoradas. Como sugerido, melhoramos o processo de criação de questões. Anteriormente o modelo LLM era usado apenas como um "filtro" inteligente, onde ele gerava questões baseado no tema e nos documentos enviados, sem usar de seu "poder natural". Na nova versão o modelo além de ler os documentos ele interpreta e usa de seu conhecimento para criar questões mais elaboradas e concisas.

Outra alteração importante foi na criação da base de dados. Agora, durante a vetorização, a aplicação verifica se o documento enviado já foi vetorizado anteriormente e evita duplicação, o que poupa tempo e recurso computacional, já que cada vetorização utiliza de tokens.

Em termos de Front-end, o projeto passou por uma leve reestruturação das telas e das paletas de cores escolhidas. A versão atual conta com o vermelho, azul e branco como cores principais, o que tornou o ambiente virtual visualmente confortável.

4.2 Melhorias no Prompt

Na fase atual do projeto, reconhecemos mais uma vez a importância crucial de um system prompt bem definido. Ele não apenas guia o que o modelo pode fazer, mas também estabelece limites claros, o que resulta em respostas muito mais precisas e satisfatórias.

A maior parte das melhorias no código está relacionada ao processo de geração de questões, que é diretamente influenciado pelo *system prompt*. Algumas das definições que nos permitiram alcançar questões de melhor qualidade são:

- Usar o contexto como apoio, não como limitação, permitindo que o modelo explore conceitos sem ficar preso ao texto original;
- Impedir a criação de questões sobre temas que não estejam nos documentos, garantindo coerência e relevância;
- Utilizar os documentos como referência conceitual, não como fonte para cópia ou citação direta;
- Priorizar questões que avaliem raciocínio, interpretação e aplicação prática, indo além da simples memorização.

Todos esses pontos, quando incorporados ao prompt do modelo, demonstraram um impacto significativo na qualidade das questões geradas tornando-as mais claras, concisas e alinhadas com os objetivos de aprendizagem.

4.3 Decisões finais

Algumas decisões finais que envolveram a entrega final foram:

1. **Front:** A paleta de cores da última versão foi mantida com leves alterações. Como recomendado, quando uma questão é selecionada, sua cor de destaque é verde e não mais vermelho, para evitar confusões. Além disso, foram implementadas estados de loading antes

da geração de questões e vetorização do documento (Antes o usuário apenas esperava a aplicação retornar às questões sem nenhum feedback visual de andamento)

2. **Back:** A arquitetura geral foi mantida e houveram algumas melhorias, como dito anteriormente, o processo de criação da base de dados foi melhorado, para evitar a duplicação de documentos, o endpoint de geração de questões também foi melhorado e, por fim, foram adicionados endpoints de substituição de questão, geração de PDF e avaliação final do exame
3. **Modelo:** Mantivemos o gemini-2.5-flash por retornar respostas satisfatórias para nossos objetivos.
4. **Gerenciamento do projeto:** Devido a dificuldades de realizar encontros semanalmente com todos os membros, as reuniões com todos os membros passaram a ser realizadas quinzenalmente, porém foram realizadas reuniões menores com menos membros para discutir pontos importantes. A metodologia utilizada ainda foi a Kaban.

4.4 Dificuldades

As nossas principais dificuldades nessa entrega foram:

1. **Limitação de tokens ao testar arquivos grandes:** Durante os testes com LLMs utilizando o Gemini, arquivos extensos não puderam ser processados integralmente, pois o limite de tokens da API era excedido. Isso dificultou a validação completa dos cenários e exigiu fragmentar os dados ou reduzir o conteúdo dos testes.
2. **Problemas de compatibilidade do backend:** O backend só funcionou corretamente para os testes após a instalação do Python 3.11. Em versões anteriores do Python, a aplicação não rodava devido a incompatibilidades com dependências, impedindo o avanço até que o ambiente fosse ajustado.
3. **Backend inicial criado sem ambiente virtual (venv):** A primeira versão do backend foi configurada sem o uso de um ambiente virtual, o que gerou conflitos de dependências e instalou pacotes diretamente no sistema. Isso dificultou a replicação e manutenção do ambiente até que o projeto fosse migrado para um venv adequado.
4. **Instalação demorada das dependências:** Devido à falta de compatibilidade entre algumas versões de pacotes e versões do Python, a instalação das dependências demorou bastante. O Python precisava testar e conciliar automaticamente quais versões funcionavam entre si, tornando o setup mais longo do que o esperado.

5. Próximos passos - Além da cadeira de EX III

Caso haja interesse de dar sequência ao projeto após o fim da cadeira, ainda há alguns pontos que não tivemos tempo suficiente para desenvolver que podem aprimorar a experiência do usuário com o ExamForge e tornar a aplicação mais autêntica e completa:

- Geração de questões subjetivas;
- Instruções detalhadas de como as questões devem ser geradas. Ex.: “Gere 3 questões de múltipla escolha sobre Redes e 2 subjetivas sobre Programação Orientada a Objetos”;
- Possibilidade de compartilhamento de exames, com vários usuários fazendo o mesmo exame simultaneamente;
- Base de dados contendo os exames feitos e os resultados.

6. Considerações finais

Este projeto representou uma experiência profundamente enriquecedora para toda a equipe. Ao longo do desenvolvimento, não apenas aplicamos conceitos técnicos, como também assimilamos práticas essenciais de gestão de projetos, colaboração e solução de problemas em um ambiente real de trabalho.

Do ponto de vista técnico, aprimoramos nossas habilidades em desenvolvimento Full Stack, utilizando ferramentas modernas de Back-end e Front-end, e fortalecemos nosso domínio sobre sistemas de controle de versão como Git e GitHub. Essas competências foram fundamentais para a construção de uma aplicação funcional e bem estruturada.

Um dos eixos centrais de aprendizado foi a exploração de Large Language Models (LLMs) e da arquitetura RAG (Retrieval-Augmented Generation). Essa imersão nos permitiu entender na prática como a IA generativa pode ser utilizada para criar soluções contextualizadas e inteligentes, conhecimentos que se mostrarão valiosíssimos tanto na carreira profissional quanto em futuras pesquisas acadêmicas.

Por fim, registramos nosso agradecimento ao professor orientador, cuja mentoria foi decisiva para o sucesso do projeto, e aos avaliadores, cujas observações e feedback são muito valorizados pela equipe. Foi uma trajetória de grande crescimento técnico e colaborativo.