Inteligência Artificial 2

Aula 30: Demoday e Pitch Final

Prof. Aléssio Miranda Júnior
alessio@cefetmg.br

CEFET-MG - Campus Timoteo
Dep. Engenharia de Computação

Inteligência Artificial 2

Sumário

Onde Paramos?

Ontem vocês publicaram o produto na **nuvem**:

- URL pública acessível por qualquer navegador.
- Variáveis de ambiente seguras no painel da plataforma.
- Pitch esboçado: Dor $\rightarrow$ Solução $\rightarrow$ Demo $\rightarrow$ Engenharia.

A Missão Final

Hoje é o **Demoday**. 10 minutos por equipe.
Provem o valor do que construíram.

Objetivos do Demoday

1. **Apresentar** o Pitch completo: problema, solução, demo ao vivo, engenharia.
2. **Demonstrar** o produto na URL pública (sem localhost!).
3. **Defender** decisões arquiteturais com dados e trade-offs.
4. **Responder** perguntas técnicas do professor e da turma.
5. **Celebrar** 30 aulas de jornada: do Perceptron ao produto na nuvem.

A Cultura do Vale do Silício

No ecossistema global de Inovação e Aceleração de Startups (ex: *Y Combinator*), os investimentos não são decididos lendo milhares de linhas de código num *pendrive*. Eles são decididos no **Demoday (Dia da Demonstração)**:

- As equipes fundadoras têm de 3 a 5 minutos para subir ao palco.
- Devem provar a tese tecnológica e o domínio sobre a Arquitetura em tempo real.
- O foco não é "O quão difícil foi programar", mas sim "Qual problema real essa tecnologia resolve?".

Soft Skills na Engenharia Moderna

Há um mito acadêmico letal: *"Meu código é tão limpo e complexo que ele se vende sozinho."* O mercado exige **Engenheiros(as) AI-First** com Soft Skills afiadas:

- **Tradução de Complexidade:** Capacidade de explicar tensores e bancos vetoriais para um Diretor que não possui *background* técnico.
- **Argumentação Baseada em Dados:** Não basta dizer que o *FastAPI* foi escolhido "porque Python é legal". Deve-se justificar pela compatibilidade com ecossistemas como *LangChain* ou suporte asíncrono.

O Desafio: Explicando RAG para Leigos

Durante o *Pitch*, vocês precisarão explicar como a IA sabe do documento privado de vocês sem que vocês a tenham re-treinado. **A explicação matemática (Não usar hoje):** *"O Sentence-Transformer mapeou as strings em um espaço n-dimensional com $n = 768$, e executamos HNSW para otimizar a matriz de similaridade de cosseno..."* **A explicação de Negócios (Usem essa!):** *"Nós dividimos os manuais da empresa em pequenos blocos de significado. Quando o cliente pergunta algo, o banco cruza os significados em frações de segundo, encontra a página exata e força a IA a ler apenas essa página antes de responder, blindando contra alucinações."*

Explicando Embeddings e Modelos Semânticos

O avaliador de negócios pode perguntar: *"Por que a busca de vocês é melhor que o Crtl+F?"*
A explicação deve ser visual e focada no resultado:

- "O *Ctrl+F* é uma busca sintática. Se você buscar por 'cachorro', ele não achará o texto se estiver escrito 'cão'."
- "A nossa ferramenta (Vector Database) converte os textos em conceitos matemáticos. Para o computador, 'cachorro' e 'cão' ocupam a mesma coordenada espacial. Nossa busca é por *intenção*, não por *letras*."

Justificativa Arquitetural: Fine-Tuning vs RAG

O avaliador mais técnico indagará as escolhas sistêmicas. **A Defesa do Custo vs Benefício:**

- *"Por que RAG e não Fine-Tuning de LoRA?"*
- "O ajuste fino custaria centenas de dólares em GPUs na nuvem para cada nova versão de conhecimento do modelo, e não garante ausência de alucinações."
- "A arquitetura RAG permite atualizarmos o banco injetando novos PDFs da empresa em milissegundos a custo quase zero, além de fornecer rastreabilidade imediata (Citações de Fontes)."

A Dinâmica de Tempo de Palco (Timeboxing)

Alerte sua equipe: o maior crime em um Demoday é estourar o tempo antes de mostrar a *Live Demo*.

- **A Síndrome do Desenvolvedor:** Passar 80% do tempo explicando os bastidores do código e 20% mostrando o produto.
- **A Regra de Ouro (Proporção 30/70):** O negócio e o mercado interessam mais. Gaste 30% na arquitetura/problema, e 70% na fluidez do produto operando ao vivo.
- Ninguém investe na complexidade do seu código Python, investem na **experiência** que o usuário final terá na ponta da cadeia.

O Custo Cognitivo (Por que a UX vence?)

Investidores (e professores) avaliam o **Custo Cognitivo**.

- Se o seu sistema de IA demanda que o usuário escreva um Prompt complexo e matemático para funcionar, o produto falhou.
- A interface (UI/UX) deve fazer a transição invisível do pensamento do cliente para a máquina.
- *"Nosso produto possui apenas um botão de upload e um campo de texto natural. Toda a Engenharia de Prompt complexa e as formatações de JSON ocorrem internamente no Backend, ocultas do cliente."*

A Importância da Equipe Multifuncional

Um projeto de IA vencedor não é composto apenas por pesquisadores de Modelos de Linguagem. Durante as rodadas de perguntas (*Q&A*):

- Direcionem as perguntas de Interface e *WebSockets* ao **Frontend Engineer**.
- Direcionem as perguntas de Banco de Dados, AWS, e CORS ao **Backend/DevOps**.
- Direcionem as métricas de RAG (ROUGE, Chunk Size) ao **Engenheiro de Dados/IA**.
- Isso atesta a maturidade corporativa da equipe em lidar com divisões de responsabilidades modernas.

A Ética e Responsabilidade Algorítmica

A Inteligência Artificial amplifica vieses. Em apresentações de alto nível, vocês devem citar a camada ética do projeto voluntariamente.

- Se o projeto envolve currículos (*RH*), qual a garantia de que a IA não excluirá candidatas femininas?
- Se o projeto lida com saúde, há um *Disclaimer* óbvio na tela alertando sobre diagnóstico médico?
- **Resposta Esperada:** "Injetamos instruções estritas no *System Prompt* de nosso LangChain proibindo o modelo de atuar como tomador de decisão final em casos que ferem os Direitos Humanos, exigindo supervisão humana (HITL)."

A Visão de Longo Prazo (Roadmap Futuro)

Projetos encerram disciplinas, mas Startups sobrevivem anos. Ao final do *Pitch*, demonstrem clareza sobre o que ainda **falta** fazer:

- *"Para os próximos 6 meses, nosso Roadmap inclui migrar da busca vetorial simples para Hybrid Search (BM25), além de implementar Autenticação via OAuth2."*
- Assumir dívida técnica publicamente é uma virtude de engenheiros experientes. Mostra que a equipe compreende a vastidão do ecossistema tecnológico, em vez de acreditar que o software atual já está "perfeito".

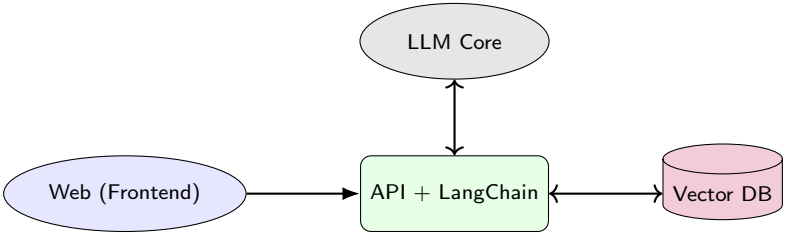
Fase 1: O Gancho e a Dor

Nenhuma arquitetura se sustenta se não curar uma "Dor". Comecem a apresentação pelo **Domínio**.

- "O Brasil possui milhões de processos judiciais parados. Advogados recém-formados gastam 4 horas lendo jurisprudências antigas."
- "O custo de atendimento de N1 é a maior sangria financeira da área de Suporte hoje."
- Deixem a sala ciente do abismo, para só depois apresentarem a IA como a ponte construída.

Fase 2: O Diagrama do Produto (System Design)

Apresentem de forma macro **como** a Inteligência foi orquestrada. Usem o conceito de blocos visuais.



"A interface roda isolada na ponta do usuário. O Backend atua como cérebro central, roteando a necessidade do usuário para o LLM e recuperando informações vitais do Banco de Vetores."

Fase 3: A Prova de Fogo (Live Demo)

Este é o momento crucial. Sem simulações ou *Mocks* em código falso. Sem vídeos gravados na noite anterior.

- **A Regra Base:** Mostrar a URL pública acessível no navegador. Sem rodar `localhost`.
- Façam o fluxo de ponta a ponta na frente da sala: *Logar (opcional)* $\rightarrow$ *Fazer o Upload do Arquivo* $\rightarrow$ *Fazer perguntas impossíveis para buscas convencionais* $\rightarrow$ *Exibir a geração RAG real.*

Lidando com Falhas ao Vivo

A Internet do auditório oscilará, a OpenAI cairá ou o Servidor Free Tier entrará em colapso. O *Demo* vai quebrar. **Como o(a) Engenheiro(a) Sênior reage?**

- Nunca minta dizendo "Ah, é um bug visual rapidinho".
- A honestidade técnica brilha: "O tempo limite (Timeout) excedeu porque o contêiner Serverless na nuvem sofreu um *Cold Start* (Inicialização fria). Se atualizarmos, a RAM estará aquecida."
- O bom engenheiro não foge do erro no terminal, ele explica o porquê o erro aconteceu.

Fase 4: Governança, Código e Custos

Para encerrar a apresentação da equipe, a demonstração da estrutura interna.

- **Gestão do Repositório:** Mostrar a aba do GitHub da equipe, as branches separadas (Frontend/Backend) e o arquivo `README.md` limpo e instrutivo.
- **LLMOps na Prática:** Mostrar que o projeto de vocês possui injeções seguras (*Environment Variables*) na nuvem e o código no Git não vazou chaves.
- **Métricas e Guardrails:** Qual a instrução de Prompt usada para impedir que a IA forneça os dados privados da empresa ou seja corrompida (Jailbreak)?

A Jornada: Do Perceptron aos Agentes

Lembrem-se das primeiras aulas do Semestre:

- Computamos $y = f(\Sigma(X \cdot W) + B)$ manualmente no papel, utilizando funções *Sigmoid* limitadas a saídas binárias 0 ou 1.
- Escalamos para Deep Learning e Backpropagation em PyTorch.
- Estudamos a Revolução dos **Transformers (2017)**, o mecanismo de *Attention is All You Need* e a ascensão dos LLMs massivos (GPT/Gemini).
- Entramos nas entranhas corporativas dos Embeddings, do RAG, de Vetores Densos e Esparsos.

A Mudança de Paradigma (Software 1.0 vs 2.0)

A IA Generativa mudou a matriz da computação.

Até 2022, criávamos as regras fixas de negócio (*If/Else*) e o programa obedecia às cegas. **Software 1.0 (Clássico)**. Culminamos na criação do **Software 2.0**: O desenvolvedor apenas

fornece a capacidade (Function Calling / Tools). O programa age como um "Gerente Terceirizado", analisando requisições humanas dúbias em linguagem natural e decidindo autonomamente (ReAct) a próxima rota a acionar.

O Futuro: Engenheiros Full-Stack AI

O programador puramente focado em centrar Divs ou criar CRUDs básicos de SQL tem os dias contados. O mercado mudou de escopo.

- Profissionais que orquestram **Interfaces, Bancos de Dados, Nuvem e Modelos Neurais** conjuntamente serão o padrão dominante da próxima década.
- Dominar APIs e arquiteturas RAG garante que vocês resolvam dores corporativas reais e agreguem valor na tomada de decisões em velocidade e em escala.

A Rubrica de Avaliação do Projeto

Como o professor, os convidados ou a banca pontuarão cada Startup hoje?

1. **Densidade e Integração (30%):** O Frontend não congela, há uso correto de rotas API e o Banco Vetorial interage com o Banco SQL fluentemente.
2. **Qualidade Cognitiva (25%):** A IA não sofre de amnésia e não alucina. Os hiperparâmetros de Ingestão de *Chunking* e do Prompt de Sistema blindam as falhas.
3. **Disponibilidade e Nuvem (25%):** O projeto **roda na internet** via Serverless/Docker sem exibir o fatídico *Error 500* ou vazamento de Segredos de Infraestrutura.
4. **Comunicação e Defesa (20%):** A equipe soube traduzir a arquitetura ao palco, vendeu a ideia com um problema palpável e justificou a stack adotada.

Hackathon Demoday

Esperamos que a disciplina de Inteligência Artificial 2 tenha desmistificado a "mágica" da IA. Na essência, não há milagres, mas sim bilhões de pesos numéricos bem otimizados, orquestrados pela mais bela Engenharia de Software.

O projeto final inicia agora. Que venham os Pitchs!

Dúvidas?

Email: alessio@cefetmg.br
Aula: Segunda e Quarta-Feira, 7h as 8:40h (Sala 39)
WhatsApp: Dúvidas rápidas

Lembrete

Confira sua Issue no GitLab!

Próxima aula: **Demoday Capstone**

100% da Aula 30 Concluída!