

Capítulo 30 – Demoday e Pitch Final

Aléssio Miranda Jr.

Novembro - 2026/2

Capítulo em Revisão

Este conteúdo está sendo revisado. Algumas informações podem estar preliminares, conter erros de formatação ou sofrer alterações na versão final.

O que você verá neste capítulo

Todo o raciocínio matemático, a engenharia de prompts, e a construção de microsserviços culminam neste momento final: o **Demoday**. Neste capítulo, discutiremos a essência do “Pitch” de startups, a vitalidade das *Soft Skills* na engenharia corporativa, e a estrutura necessária para provar, ao vivo e sob pressão, o valor do sistema de Inteligência Artificial que a sua equipe construiu ao longo das últimas aulas.

1 O Que é o Demoday?

O termo *Demoday* (Dia de Demonstração) nasceu e se cristalizou no agressivo ecossistema de Startups e Aceleradoras do Vale do Silício, popularizado por instituições globais como a *Y Combinator*. Após meses de desenvolvimento intenso trancados em escritórios, sobrevivendo de financiamento-anjo e consumindo horas insones de codificação, as equipes fundadoras ganham a chance de subir a um palco. Em meros cinco a dez minutos, eles devem provar o valor comercial, a robustez arquitetural e o impacto de suas tecnologias para um painel impiedoso de jurados e potenciais investidores de *Venture Capital*.

Na nossa disciplina, o **Demoday Capstone** representa a prova e a defesa oral máxima de todos os conhecimentos absorvidos no semestre. É a validação que une o primeiro laboratório sobre derivadas e neurônios artificiais ao uso de bancos de dados vetoriais, LLMOps e RAG distribuído na nuvem.

2 Soft Skills na Engenharia de Software Moderna

Existe uma miopia romântica e perigosa na Ciência da Computação: a de que “um bom código vende a si mesmo” ou “se o produto for brilhante matematicamente, o mercado vai comprar”. Na realidade brutal do mundo corporativo moderno, o Engenheiro de Software ou o Arquiteto de Machine Learning não programa isolado em um porão.

Para escalar a carreira, além do domínio da IDE (*Hard Skills*), você precisa ter **Soft Skills** afiadíssimas. Você precisará ser capaz de:

• Teoria: title=As Duas Grandes Habilidades de Comunicação

- **A Tradução da Complexidade:** Você deve conseguir olhar nos olhos de um diretor financeiro ou de um CEO (que não sabe o que é cálculo tensorial, vetor ou JSON) e explicar *exatamente* como um Banco de Vetores economiza dinheiro e resolve o problema de esquecimento da inteligência artificial.
- **A Defesa Arquitetural:** Você será desafiado por outros engenheiros do time. Você deve justificar a escolha de suas ferramentas com base em três pilares pragmáticos: *Custo*, *Latência* (tempo de resposta) e *Manutenibilidade*. Por que sua equipe usou RAG em vez de Fine-Tuning? Por que Pinecone em vez de ChromaDB? O “porque eu acho legal” não existe em engenharia.

3 A Estrutura do Pitch

Para a avaliação final da disciplina, espera-se que a apresentação da equipe flua seguindo a lógica do funil de produto. Não subam ao palco para focar exclusivamente na tela do VSCode, exibindo funções if/else intermináveis. A demonstração é técnica, mas focada na **solução**.

A estrutura recomendada do Pitch engloba os seguintes quatro pilares:

► Prática: title=Roteiro do Demoday Capstone

- 1 A Dor e o Problema:** O que o mundo, ou o setor de um cliente específico, sofre hoje que vocês decidiram resolver? Qual o desperdício de tempo ou dinheiro atual?
- 2 O Diagrama de Solução Tecnológica:** Qual é o fluxo de IA implementado? Neste momento da apresentação, deve-se mostrar um mapa visual do fluxo (ex: Usuário envia PDF → Backend Python quebra o texto → Criação de Embeddings → Gravação no Pinecone → Rotação RAG).
- 3 A Prova ao Vivo (Live Demo):** Uso obrigatório do produto rodando na URL pública oficial da equipe. Nada de *localhost*. Façam uma requisição real para que a banca presencie a resposta da inteligência gerada na nuvem na frente de todos.
- 4 A Qualidade de Engenharia (Clean Code):** Uma passagem pontual e rápida sobre a organização da estrutura do repositório no GitLab, evidenciando divisão lógica de responsabilidades, ausência letal de chaves soltas no arquivo `main` e legibilidade de código.

Tabela 1: Gestão do Tempo no Pitch (Exemplo para 10 Minutos)

Seção	Duração	Objetivo do Orador
1. A Dor	2 min	Capturar a atenção da banca com dados reais do problema.
2. A Solução	2 min	Explicar a arquitetura RAG/Agentes de forma visual e clara.
3. Live Demo	4 min	Mostrar a interface real operando perfeitamente online.
4. Engenharia	2 min	Demonstrar repositório estruturado, commits e esteira CI/CD.

4 Encerramento: A Fronteira Alcançada

A apresentação triunfal no Demoday não é apenas o cumprimento de uma nota acadêmica; é o atestado material da sua evolução como engenheiro. Vocês entraram nesta disciplina utilizando simples lógicas condicionais e scripts lineares no terminal para processar CSVs. Finalizam o arco construindo produtos arquitetados e conectados na internet, enraizados na tecnologia computacional mais profunda e disruptiva desta década.

Nós superamos o ceticismo inicial e as barreiras do código duro. Parabéns pelo avanço fenomenal e por se consolidarem como Arquitetos e Engenheiros em Inteligência Artificial. O mundo lá fora espera o próximo grande *Commit* de vocês.

✓ Takeaways

- O **Demoday** é a validação final que une teoria e prática: do primeiro neurônio artificial ao RAG distribuído na nuvem.
- **Soft Skills** são tão importantes quanto Hard Skills: a capacidade de traduzir complexidade técnica para não-técnicos e defender decisões arquiteturais com dados separa engenheiros medianos de líderes.
- O Pitch segue um funil de 4 pilares: **Dor** (o problema) → **Solução** (diagrama da arquitetura) → **Live Demo** (prova ao vivo na URL pública) → **Engenharia** (repositório estruturado).
- A gestão do tempo é crítica: 2 min para a Dor, 2 min para a Solução, 4 min para o Demo, 2 min para a Engenharia — não há espaço para improviso.
- “Porque eu acho legal” não é justificativa em engenharia. Decisões devem ser baseadas em **Custo**, **Latência** e **Manutenibilidade**.

Questões: Autoavaliação

- 1 **[Direta]** Descreva os 4 pilares do roteiro de Pitch (Dor, Solução, Live Demo, Engenharia) e explique o objetivo comunicacional de cada etapa.
- 2 **[Direta]** Por que a Live Demo deve rodar em URL pública e não em localhost? Que impressão o localhost causa na banca avaliadora?
- 3 **[Direta]** Cite as duas “grandes habilidades de comunicação” descritas no capítulo e dê um exemplo prático de cada uma no contexto de uma startup de IA.
- 4 **[Reflexiva]** Reflita sobre a jornada completa da disciplina: de multiplicação de matrizes NumPy (Aula 1) a Agentes Autônomos e Deploy na Nuvem (Aula 29). Qual conceito teve o maior impacto na sua visão sobre Engenharia de IA e por quê?
- 5 **[Reflexiva]** Discuta a tensão entre “código que funciona” e “código que impressiona na apresentação”. Em um cenário de Hackathon com tempo limitado, como equilibrar qualidade técnica e impacto visual?

Lab. 30: Demoday

Capítulo em Revisão

Este conteúdo está sendo revisado. Algumas informações podem estar preliminares, conter erros de formatação ou sofrer alterações na versão final.

★ Objetivo do Laboratório

Não haverá programação intensiva nesta aula. O foco é a comunicação e a defesa do software. O laboratório se concentra nas apresentações das equipes (*Pitche*) para o painel de avaliação.

5 Atividade Prática

1. A Apresentação Oral

Cada equipe disporá de **10 a 15 minutos** para se apresentar à turma. Pelo menos dois membros (ou todos) devem falar. Organize a pauta:

- **1 minuto:** Apresentação da equipe e do nicho (Ex: “Somos a equipe LexIA e resolvemos análise de contratos”).
- **3 minutos:** Arquitetura. Qual linguagem, framework (LangChain, LlamaIndex ou Spring AI), API escolhida e Banco de Vetores. Mostre rapidamente um diagrama de fluxo ou a arquitetura em tópicos.
- **5 minutos:** O *Showcase* (Demo). Abra a URL pública do projeto de vocês na frente da sala. Mostre a aplicação falhando caso uma regra não seja atendida e a aplicação retornando o conhecimento caso as informações estejam corretas (evidenciando a inteligência conectada).
- **Restante:** Mostre a organização do Git (commits, estrutura, Readme) como comprovante do trabalho executado ao longo do Hackathon.

2. Argumento

Os jurados (Professor(es) ou colegas da sala) poderão fazer de 1 a 2 perguntas técnicas focadas nas decisões arquiteturais do time. Exemplo: “Se o volume de documentos duplicar amanhã, o que precisariam mudar na arquitetura de vocês?”. Respondam embasados nos conhecimentos das aulas de IA e RAG da disciplina.

6 Critério de Avaliação Final

A nota da equipe no Trabalho Final englobará a qualidade do sistema em produção (estabilidade das requisições na nuvem), as métricas de desenvolvimento contínuo (Commits não-centralizados em apenas um aluno no GitLab) e o nível de clareza demonstrado pela dupla/trio durante a defesa do Demoday na explicação das abstrações de inteligência artificial.

✓ Takeaways

- Vocês construíram um **produto real de IA de ponta a ponta**: da teoria do neurônio artificial (Aula 1) ao RAG corporativo na nuvem com interface visual (Aula 30).
- O Pitch validou duas habilidades essenciais: **traduzir complexidade técnica** para não-técnicos e **defender decisões arquiteturais** com dados.
- O Git conta a história: os commits mostram quem fez o quê, quando e por quê — transparência total.
- Vocês são, oficialmente, **Engenheiros de Sistemas Inteligentes**. O próximo grande commit é de vocês.