MINDUS DATA SCHOOL

GUIA PRÁTICO · 2026

Anti-vitrine

O projeto que sobrevive à entrevista

Sete etapas, as perguntas que o entrevistador faz em cada uma, e como escrever tudo isso no seu perfil sem parecer lista de ferramenta.

AS TRÊS PERGUNTAS QUE DERRUBAM UM PROJETO DE VITRINE

01
Quanto custava rodar isso?

02
O que quebrou, e como você descobriu?

03
Por que essa escolha e não a outra?



José Lucas Vasconcelos · Senior Analytics Engineer
school.mindusdata.com.br

POR QUE SEU PROJETO NÃO CONVERTE

Projeto de vitrine é o que existe pra ser visto, não pra resolver.

Ele é bonito. Dashboard colorido, stack da moda, README caprichado. O problema aparece na entrevista, quando alguém pergunta o que ele resolvia.

Quase todo projeto de portfólio morre na segunda pergunta. Não porque a pessoa não sabe a ferramenta, mas porque ela construiu de trás pra frente: escolheu a stack primeiro e foi procurar um dataset que coubesse nela.

Quem constrói pra resolver faz o caminho inverso, e por isso responde sem pensar.

Vitrine

- Começa escolhendo a ferramenta
- Dataset limpo, baixado pronto
- Roda no notebook, quando você manda
- Nunca quebrou
- Não tem custo, porque nunca subiu
- Descrito por lista de tecnologias

Anti-vitrine

- Começa escolhendo o problema
- Fonte suja, que muda e falha
- Roda sozinho, em horário definido
- Já quebrou, e você sabe contar como
- Você sabe a conta do mês
- Descrito pelo que decidiu

Nenhum item da coluna da direita exige ferramenta mais cara. Todos exigem uma decisão a mais.

A regra que filtra tudo

Se você não consegue dizer **quem perde dinheiro ou tempo** quando esse dado falta, não é um projeto. É um exercício. E exercício não passa em entrevista.

AS SETE ETAPAS

Um projeto só. Sete lugares onde ele pode virar vitrine.

Cada etapa carrega uma pergunta que o entrevistador faz. Se você tem a resposta com número, a etapa está feita. Se a resposta é "não cheguei a ver isso", achou onde trabalhar.

1. **Fonte suja**
   "De onde vinha o dado e o que ele tinha de errado?"
2. **Ingestão**
   "O que acontece se rodar duas vezes?"
3. **Armazenamento**
   "Por que esse banco e não outro? Quanto custa?"
4. **Transformação**
   "Como você sabe que o número está certo?"
5. **Orquestração**
   "Como você fica sabendo que quebrou?"
6. **Produto final**
   "Quem usa isso e pra decidir o quê?"
7. **Deploy e custo**
   "Quanto custa por mês, e como você cortaria pela metade?"

01 Fonte suja

Não use CSV limpo de repositório de exercício. Use algo que resista: uma API pública com paginação e limite de requisição, uma planilha que alguém edita à mão, um site que muda o layout. O valor do projeto começa aqui, porque é aqui que ele passa a ter defeito.

> **A PERGUNTA**
> De onde vinha o dado e o que ele tinha de errado?

Sinal de vitrine: "peguei um dataset pronto".

02 Ingestão

Um script que busca e grava o dado **bruto**, sem transformar nada. Guarde o payload original: quando o número der errado seis meses depois, é ele que salva você. Duas propriedades valem mais que o resto: rodar duas vezes não pode duplicar, e cair no meio não pode deixar lixo pela metade.

> **A PERGUNTA**
> O que acontece se rodar duas vezes? E se a API cair no meio da execução?

Sinal de vitrine: o script só funciona se tudo der certo.

03 Armazenamento

Duas camadas no mínimo: o bruto preservado e o tratado. Postgres, DuckDB ou Parquet em object storage resolvem a grande maioria dos projetos de portfólio, e custam perto de nada. O que importa não é qual você escolheu, é você conseguir dizer por quê.

> **A PERGUNTA**
> Por que esse banco e não outro? Quanto ele custa por mês?

Resposta que derruba: "era o que eu já conhecia". Resposta que segura: volume, padrão de consulta e custo.

04 Transformação

SQL versionado, em camadas, com teste. Se você usa dbt, os quatro testes de prateleira já te colocam à frente da maioria: **not_null**, **unique**, **accepted_values** e **relationships**. Eles são baratos de escrever e são a única prova objetiva de que o seu número não é chute.

> A PERGUNTA
>
> Como você sabe que o número está certo?

Sinal de vitrine: "eu conferi na mão uma vez".

05 Orquestração e alerta

O projeto roda sozinho, em horário definido, sem o seu notebook ligado. GitHub Actions agendado resolve de graça. E, mais importante que rodar: quando falhar, alguém precisa ficar sabendo. Um webhook no Discord ou um e-mail já basta.

Essa é a diferença entre um script e um sistema. Sistema avisa quando se machuca.

> A PERGUNTA
>
> Como você fica sabendo que quebrou?

Sinal de vitrine: "eu rodo quando preciso".

Falha silenciosa é sempre a mais cara

O dado não chega, ninguém reclama, o dashboard continua verde, e três semanas depois alguém pergunta por que o número está estranho. Nesse ponto você já tomou decisão errada por três semanas. É por isso que a etapa 5 vale mais do que parece.

06 Produto final

Alguém decide alguma coisa olhando o que você construiu. Pode ser um painel, um alerta ou um endpoint. O formato importa menos que existir um destinatário.

> **A PERGUNTA**
>
> Quem usa isso e pra decidir o quê?

07 Deploy e custo

Está no ar. Funciona com sua máquina desligada, e você sabe a conta do mês, mesmo que ela seja zero. Saber o custo é o que separa quem fez um exercício de quem operou alguma coisa.

> **A PERGUNTA**
>
> Quanto custa por mês? E se eu pedisse pra cortar pela metade, o que você faria?

A ETAPA QUE QUASE NINGUÉM TEM

Quebre o seu projeto de propósito.

Injete um registro ruim de propósito e deixe o pipeline bater nele. Data no futuro. Id duplicado. Campo obrigatório nulo. Acentuação quebrada. Um número que veio como texto.

Depois documente três coisas, em três linhas:

- → Qual teste pegou, ou quanto tempo levou até você perceber
- → O que você mudou pra não acontecer de novo
- → O que você deixou passar de propósito, e por quê

Isso vira a sua melhor resposta de entrevista. Todo candidato conta o que construiu. Quase nenhum consegue contar o que quebrou e como descobriu, porque nunca esteve perto de um sistema que quebra.

PARTE 2 · POSICIONAMENTO

Construir é metade. A outra metade é saber escrever.

Dá pra ter o melhor projeto da sua turma e ele não passar do primeiro filtro, porque o recrutador tem oito segundos e o que ele vê é uma lista de tecnologias que ele não sabe avaliar.

A correção leva dez minutos e o resultado aparece no mesmo dia.

A FÓRMULA DE TRÊS LINHAS

1. **O problema**, em dinheiro ou em tempo.
   Não a ferramenta. O prejuízo.
2. **O que você construiu**, com a stack entre parênteses.
   A stack vira nota de rodapé, não manchete.
3. **O trade-off que você defende**, ou o custo.
   É a linha que prova que teve decisão, e não tutorial.

O MESMO PROJETO, ESCRITO DE DOIS JEITOS

A diferença não está no que você fez. Está no que você contou.

ANTES

Projeto de Engenharia de Dados | Python, Airflow, dbt, Snowflake e Power BI. Pipeline completo de ingestão, tratamento e visualização de dados.
#dataengineering #python #dados #dbt

Diz o que você usou. Não diz o que você resolveu, pra quem, nem o que custou. O recrutador não tem como avaliar nada disso.

DEPOIS

O estoque do site ficava até 6 horas atrasado em relação ao ERP, e a loja vendia produto que não tinha.

Construí um pipeline que sincroniza a cada 15 minutos e avisa no Discord quando falha (Python + dbt + Postgres, agendado no GitHub Actions, custo de R$ 0 por mês).

Escolhi Postgres em vez de um data warehouse pago porque o volume é de 200 mil linhas por dia e o custo fixo não se justificava.

Problema, número, decisão. Quem lê sabe te avaliar, e já tem três ganchos pra te perguntar na entrevista. Os ganchos são seus: você escolheu quais.

Onde isso vai

- → Na seção **Projetos** do LinkedIn, como descrição
- → Nas três primeiras linhas do **README** do repositório, antes de qualquer instrução de instalação
- → Na sua **seção Sobre**, em versão de uma frase

ANTES DE PUBLICAR

O teste dos 30 segundos

Leia a descrição do seu projeto em voz alta, cronometrando. Se em 30 segundos não aparecer as três coisas abaixo, ainda é vitrine.

1	2	3
Um problema que custa dinheiro ou tempo	Um número qualquer que seja	Uma escolha que você defende

Se faltar o problema, volte para a página 1: você começou pela ferramenta.
Se faltar o número, volte para as etapas 3 e 7: você não mediu nem custeou.
Se faltar a escolha, volte para a quebra proposital: você não tem cicatriz pra contar.

Isso aqui é um pedaço pequeno

A parte 2 deste guia é uma amostra do bloco de posicionamento da mentoria da Turma 2 da Mindus Data School, onde a gente reescreve seu LinkedIn junto e eu te entrevisto de verdade, com as perguntas que caem.

A turma abre em novembro, e quem está na lista sabe a data primeiro.
school.mindusdata.com.br