

Como raio se põe em prática uma ambição pedagógica de usar IA?

Um caso com projetos pessoais, dialética apoiada por IA e 220 estudantes em Programação por Objetos.



LEONEL MORGADO

JUL 11, 2026

🇬🇧 There's an English language version.

<https://doi.org/10.5281/zenodo.21309162>

Young Builders

Como raio se põe em prática uma ambição pedagógica de usar IA?

Projetos pessoais, dialética apoiada por IA e 220 estudantes em Programação por Objetos

Leonel Morgado

CASO REAL - 2025/26

Grças à app...

Pre-visualização

Arquétipo: Dado de
Especie: Halfling
Antecedente: Urbana-Bastarda
For Des Car 9 17 14

Licenciatura em Eng. Informática
Programação por Objetos 25/26

Problemas

Para aqui, foi feito um PDF com este documento, enviado para os GPT, e as imagens (por se à data, esses GPT não estão a conseguir e a não gerar automaticamente). O desejável é que com esse feedback me ajudem a entregar!

Para a desenvolver o semestre, quanto mais refinado agora, melhor.

Consequência	Limite aceite	Princípio OC
Alterações de estado da personagem fora da sua própria interface.	0 alterações ao estado da personagem fora da sua própria interface.	Encapsulamento
Lógica duplicada/dispersa, difícil de testar.	1 ponto único que calcula impacto antes de confirmar.	Abstração
Classes "faz-tudo", alto	1 razão para	Responsabilidade

Também outros comportamentos, à medida que discutirmos e refletirmos sobre a matéria, irão provavelmente constatar-se. Por exemplo: podemos querer "Consultar" um arquétipo, o que contém, mas é dentro do Arquétipo que o "consultamos". Não! Muito provavelmente, é na Personagem que se encontra. Ou então, é noutra parte da aplicação que o ato de "consultar" ocorre a personagem, ou seja, o mesmo poderíamos usar para "Antecedente". Provavelmente não é o antecedente em si a ser apagado, mas sim a associação de um antecedente a uma ação. Em modelação, é normal não acertarmos à primeira, pois, no entanto, precisamente porque começamos a ter alguma separação de responsabilidades entre partes do código.

Neste trabalho, é expectável que parte destes problemas seja

Há pouco tempo escrevi aqui que a IA nos obrigará a levar a sério ambições educativas que proclamamos desde o século XIX:

compreensão, propósito, apropriação pessoal, capacidade de ajuizar, decidir e assumir. São ideias bonitas. É relativamente fácil defendê-las num texto.

Depois chega a segunda-feira.

Young Builders is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.

✓ Subscribed

As ideias têm de aparecer em atividades concretas, com enunciados que os estudantes consigam interpretar, trabalho compatível com os créditos da unidade curricular. Tem de ser feito o acompanhamento (no meu caso na Universidade Aberta, assincronamente e por escrito em fóruns no Moodle). Tem de caber nos prazos. E é preciso fazer a avaliação. Isto tudo com os estudantes a usar GPT ou outras IA. Eu encorajo-os vivamente a fazê-lo, mas se os desencorajasse usariam na mesma. Podem usá-las bem, usá-las mal... usá-las com sentido crítico, para explorar uma dificuldade, ou simplesmente para produzir qualquer coisa que ocupe o espaço previsto na entrega. O trabalho do professor não deve ser balizar ingenuamente quando usar as ferramentas ou esvair-se a tentar detetar-lhes o rasto, numa quimera. Pelo menos não acho que isso seja uma forma útil de atuar profissionalmente. Interessa-me mais melhorar a ambição da aprendizagem e encontrar processos praticáveis pela generalidade dos estudantes e dos professores. Ou seja, não apenas por quem já tenha elevada motivação, empenho, resiliência e dedicação. Não! Processos que funcionem na complexa realidade dos seres humanos como são, incluindo os que provavelmente não chegariam lá sem orientação, apoio ou vivência de experiências enriquecedoras.

Este texto é sobre uma das minhas tentativas concretas. Fi-la no ano letivo de 2025/26, numa unidade curricular de Programação por Objetos da Licenciatura em Engenharia Informática da Universidade Aberta, com 220 estudantes (mais ou menos: não é estável ao longo do semestre). Não pensem que vou apresentar resultados miríficos de

ganhos de aprendizagem. Descobrir se algo é melhor do que outra coisa na educação, ou mesmo melhor que não fazer nada, é tarefa ingrata, demorada, porque não passamos da mesma maneira pelos mesmos alunos duas vezes. Saber isso demorará anos, investigação meticulosa nos planos, na execução, etc. O que tenho agora é um desenho de atividade que efetivamente foi aplicado, algo real, não apenas idealizado. Tenho os artefactos que o materializaram, observações sobre a sua execução e uma ideia bastante mais nítida do que me parece ter corrido bem e do que ambiciono fazer melhor.

O que ficava por ensinar quando ensinávamos Programação por Objetos

Para quem vem de outra área, não se assuste disto ser informática, que me parece que será útil para (quase) todos. Basta uma explicação breve de enquadramento. A programação orientada a objetos é uma forma (chamemos-lhe assim) de organização estrutural dos programas informáticos, que visa torná-los mais compreensíveis, modificáveis e fáceis de gerir à medida que crescem e começa a haver muitas mãos e cabeças a mexer-lhes, não apenas um artesão informático habilidoso. Para isso, usa várias técnicas, com nomes que refletem ideais, como “encapsulamento”, “sobrecarga”, “herança” ou “polimorfismo”. Nesta publicação não importa ao certo o que significam: importa apenas perceber que são técnicas de organização e estruturação de coisas complexas.

Na prática, grande parte do ensino, por pressões de escala (número de alunos), heterogeneidade (variedade do nível de desempenho de cada aluno) e tempo (é preciso explicar?) acabava por se concentrar na expressão e treino da sintaxe e das técnicas. Há boas razões para isso, note-se. São complexas, requerem repetição e experiência, o código tem de funcionar e os erros mais básicos podem impedir o estudante de avançar. É possível (na verdade, comum) ensinar encapsulamento sem que o estudante chegue a sentir que problema resolveu com essa

técnica. Pode implementar herança sem formar um critério sobre quando ela facilitará a evolução do programa ou antes o deixará preso a uma má decisão estrutural.

Uma solução costumeira para isto é desenvolver projetos individuais. Quando alguém tenta concretizar uma ideia que seja mesmo sua, as classes, em vez de macaquearem exemplos preparados pelo professor passam a refletir responsabilidades para com essa ideia. O diabo surge nos detalhes: há coerência no estado de cada elemento? Deixa-se que algo possa ser mudado, replicado, ou impõe-se como estático, único? Identificam-se práticas comuns e de colaboração e confiança, ou pequenas diferenças geram fronteiras e separação? (Calma, colegas informáticos, estou a escrever assim para que não informáticos possam seguir o argumento geral.) É nestes detalhes peça a peça, linha a linha, que a compreensão dos princípios se revela útil.

Esse caminho da aprendizagem baseada em projetos tem (tinha?) um problema logístico: acompanhar regularmente os projetos de uma unidade curricular com mais de duzentos estudantes, ainda por cima numa universidade online, distribuiria o *feedback* de forma inevitavelmente desigual (isto mesmo com tutores a acompanhar grupos). Alguns estudantes colocariam dúvidas mais cedo, com menos pressão docente, e poderiam receber mais atenção. Outros avançariam em silêncio durante semanas, para nunca emitirem dúvidas ou só o fazerem em catadupa, todos ao mesmo tempo. Dar a todos uma apreciação individual, detalhada e atempada exigiria um esforço incomportável da equipa docente... e já agora, dos alunos, porque se estivermos a dar *feedback* público, imaginem ver o *feedback* individual a tantos colegas!

A IA alterou este estado de coisas. Tornou viável pedir mais concretizações, criar ciclos curtos de *feedback* para todos e dedicar a atenção docente aos aspetos em que a experiência vivida, o conhecimento da turma, a leitura da pessoa e da situação, e o juízo profissional realmente acrescentam alguma coisa.

Espero que seja perceptível que este contexto não pertence apenas à informática. Em muitas áreas, é um desafio que conceitos, técnicas e princípios complexos possam ser verdadeiramente compreendidos, dominados e aplicados ("mobilizados" como dizem de maneira adorável os colegas brasileiros). A aprendizagem por projetos é comum nessas situações, sendo que este caso na programação orientada a objetos é apenas a minha circunstância pessoal.

Duas semanas com uma ideia a dar luta

O projeto do semestre não é dado pelo professor. É proposto pelo estudante, porque tem de ser o estudante a querer concretizá-lo. Se for dado pelo professor, os critérios do que interessa desaparecem, são todos uma ilusão, porque o verdadeiro critério passa a ser «agradar ao professor» ou «acertar no que o professor quer». Então, tudo começa por uma proposta do estudante, o professor só dá os contornos qualitativos. Pedi que tivesse relevância pessoal ou profissional, fosse rica em decisões de estruturação e coubesse no tempo disponível. «Fazer uma aplicação» não chega. «Fazer uma aplicação inovadora para melhorar a experiência dos utilizadores» ainda diz menos, por fingir mais. O estudante precisa de descobrir o que quer realmente pôr a nu, fazer acontecer e por que motivo isso lhe interessa.

Na primeira semana, a proposta atravessa várias formas.

Começa por apenas um parágrafo com a ideia de projeto. Curto! Nada de parágrafos saramaguianos em extensão (e habitualmente vazios de intenção, já dantes assim era, na era do GPT mais é). Essa ideia tem logo de ser concretizada numa micronarrativa de 25 a 35 palavras, produzida através de uma abordagem estruturada chamada "prototipagem de microficção científica" (PμFC, técnica que aprendi com o Vic Callaghan), que requer pensar em personagens, situações, ações. É a primeira de muitas conversões da ideia. A segunda é mudar para uma visão analítica da ideia: que entidades existem nela, que comportamentos têm essas entidades ou há entre entidades? E as

conversões cognitivas continuam: depois o estudante revela para essa ideia uma ou duas narrativas mais claras de como imagina o seu projeto a concretizar-se em situações reais: coisas ainda assim muito curtas, os informáticos chamam-lhes *user stories* (é um estrangeirismo foleiro, o pessoal provavelmente acha “mais técnico”, mas é o que se usa). Depois nova conversão da ideia!) representa-as como guiões visuais em banda desenhada (o nome chique é *storyboards*) e desenha um ou dois esboços (o termo é “*mockups*” ou “protótipos de baixa fidelidade”) de como imagina que virão a ser os ecrãs das aplicações informáticas usadas nessas narrativas.

O desenho pode ser tosco: a estética não está em avaliação. Também pode ser bonito, fica ao gosto de cada um. Interessa é perceber, nessa narrativa visual, se a situação foi mesmo imaginada e é credível e pertinente, se faz sentido face à ideia original esclarecida na micronarrativa, se os esboços de ecrãs refletem o que devem permitir que aconteça no guião visual da narrativa. A ideia, em cada uma das sucessivas transformações cognitivas, vai deixando de ter onde se esconder dentro da cabeça de cada um: as frases bem-intencionadas já deixam de fora os pés e barrigas das ideias, os adjetivos fúteis colidem com o mundo real revelado nas narrativas e esboços.

Na segunda semana, começa-se a fazer a ligação cognitiva à matéria, cujo estudo tinha sido solicitado desde o início: chega a necessidade de identificar as primeiras “classes” candidatas, um diagrama simples delas e pequenos protótipos delas em código Python. O enunciado limita propositadamente a escala do que se entrega: duas ou três classes pequenas, um ou dois métodos além do construtor (leitor: se não sabes o que é, não faz mal), duas ou três instâncias (leitor: se não souberes o que é regista “outro conceito técnico”) e cerca de sessenta linhas de código. Ou seja, o estudante não tem de entregar tudo o que faz, porque em ambiente de IA disponível o volume não demonstra valor. Tem é de escolher exemplos relevantes para os propósitos de lhe darmos *feedback* e avaliação: só precisamos de matéria suficiente para

confrontar todos os elementos entre si, para ver se são coerentes, se refletem um interesse pessoal, uma escolha pessoal, etc.

Uma ideia é obrigada a ganhar forma

O mesmo projeto atravessa representações diferentes; cada passagem expõe escolhas e incompatibilidades.



O mesmo projeto é expresso de várias maneiras. As imagens pertencem ao exemplo fornecido aos estudantes, não a trabalhos de estudantes.

É uma concretização bastante literal de algo que Seymour Papert e Idit Harel associaram ao construcionismo: construir o conhecimento enquanto se constrói uma entidade que pode ser sujeita ao “teste da realidade”: sustem-se na realidade, pode ser mostrada, discutida e reformulada. Uri Wilensky, num capítulo do mesmo livro, propôs que não há conceitos abstractos e concretos como coisas diferentes: a abstração não é uma propriedade fixa do conceito, do objeto. Depende da riqueza da relação que construímos com ela/ele: das representações, interações e ligações que conseguimos estabelecer. Uma ideia de projeto torna-se mais concreta quando o estudante lhe dá vida, a traz para o mundo real: quando a narra, desenha, modela, programa e encontra diferenças entre essas formas.

O conflito quase inevitável ao longo das transformações tem aqui uma função de revelação, de exigência de pensamento crítico, de reflexão. Recorrendo ao exemplo, se na micronarrativa da ideia uma personagem pode ter vários sítios onde guarda equipamento, mas o diagrama de classes revela que só há um inventário global, temos um conflito. Se o *storyboard* diz que o utilizador está a comparar escolhas, mas o *mockup* só revela espaço para uma, temos um conflito. Se o estudante alega que há encapsulamento (conceito da matéria) do inventário, mas o guião revela que o mestre de jogo o pode mudar diretamente, temos um conflito. Uma ideia de projeto, quer em geral que na ligação à matéria (orientação a objetos) deixa de ser uma lista de palavras. O estudante passa a ter um problema seu para deslindar, o maior dos problemas: *o que quero eu?*

Dois GPT para estruturar a dialética num ecossistema aberto

Os estudantes puderam usar quaisquer GPT ao longo da unidade curricular. Forneci-lhes também dois GPT personalizados, acessíveis por ligação direta nas páginas da unidade curricular. Não eram as ferramentas “autorizadas”, nem eram “indispensáveis”. Serviam para colocar qualquer estudante perante duas modalidades de dialética já estruturadas com os objetivos da unidade curricular, das tarefas e com os princípios de programação por objetos em estudo.

O Sócrates Digital pergunta, insiste e procura incoerências entre aquilo que o estudante afirma querer, o que propôs e as formas como começou a concretizá-lo. Deve resistir à tentação de oferecer uma solução pronta, mas sabemos que impedi-lo é impossível, simplesmente não tende a saltar para aí. O segundo GPT combina uma orientação aristotélica e estoica. Ajuda a explorar as consequências das escolhas e a levar o estudante a decidir que limites aceita para as suas ambições.

Poderiam ser três GPT: um para a incoerência, outro para a consequência, outro para os limites. Optei por dois porque a atividade tinha dois momentos. Primeiro, descobrir a distância entre a intenção e a concretização desenhada. Depois, perceber o que essa distância fará ao projeto e assumir até onde é razoável resolvê-la durante o semestre.



Os GPT fornecidos trazem contexto e uma forma de diálogo. As interpelações usadas para os configurar também são partilhadas.

Este último ponto é importante. Sim, eu sei que se não fosse não se trazia. Mas friso-o para cativar a vossa atenção. Um projeto académico não tem de ser perfeito. Nem sequer algo ser “bom” significa o mesmo para todos os projetos, todos os estudantes e todos os momentos do semestre. Há variação do tempo disponível, da experiência prévia, do risco que estamos dispostos a assumir, do esforço possível ou desejado, e à volta de tudo os conceitos da matéria da unidade curricular. Um estudante pode manter uma opção que agora é mais simples mas lhe complicará o trabalho futuro. Precisa é de constatar essa consequência, delimitar o compromisso e assumir essa decisão.

Por isso, o mapa pedido na segunda semana tinha quatro colunas:

Incoerência → Consequência → Limite aceite → Princípio de programação por objetos

MAPA DE DECISÕES

AbERTA

Licenciatura em Eng. Informática

Programação por Objetos 25/26

Mapa de decisões

Para se chegar aqui, foi feito um PDF com este documento, enviado para os GPT, juntamente com as imagens (porque à data, esses GPT não estão a conseguir extrair as imagens do PDF automaticamente). O desejável é que com esse feedback melhorem e reformulem, até à entrega!

Este será o projeto a desenvolver no semestre, quanto mais refinado agora, melhor.

Incoerência	Consequência	Limite aceite	Princípio OO
Estado da Personagem pode ser mudado por outros objetos (p.ex., Escolher() em Arquétipo/Espécie; aplicar regras "de fora").	Pontos de mutação difusos; invariantes frágeis.	0 alterações ao estado da Personagem fora da sua própria interface.	Encapsulamento; Estado/Comportamento
Comparar escolhas existe na UI mas não tem "lugar" no modelo.	Lógica duplicada/dispersa; difícil de testar.	1 ponto único que calcula impacto antes de confirmar.	Abstração
Ficha acumula papéis (guardar, coerência, exportar).	Classe "faz-tudo"; alto acoplamento.	1 razão para mudar por classe; 0 dependências cíclicas.	Responsabilidade única

SÍNTESE REFLEXIVA

AbERTA

Licenciatura em Eng. Informática

Programação por Objetos 25/26

Síntese reflexiva

Incoerência: a principal incoerência identificada, após o diálogo com o GPT Sócrático, é a diferença entre o diagrama (onde Arquétipo/Espécie/Antecedente aparecem como classes) e o protótipo em Python, onde são tratados como valores (strings). Esta escolha simplifica o código, mas fragiliza a validação e pode confundir a distinção **classe vs. instância**; além disso, baixa o nível de **abstração**, porque a representação já não acompanha o modelo conceptual indicado no diagrama.

Consequência: com o GPT Aristotélico-Estoico, conclui que a consequência prática é um ganho imediato de tempo em focar-me nas funcionalidades fundamentais da ideia, sob pena de ter um aumento imprevisível de esforço futuro caso aumente a complexidade. Por exemplo, posso ter dois sistemas de regras quase iguais, só com diferenças aritméticas básicas, para demonstrar o conceito... depois se quiser ser mais realista e ter sistemas muito diferentes, isso ir-me-á consumir muito tempo e arrisco-me a forçar violações da **responsabilidade única** (porque com a pressão posso acabar por ter regras implementadas presas por arames, espalhadas por sítios menos adequados).

Limite aceite: vou manter só um sistema de regras principal (D&D) e tratar Arquétipo/Espécie/Antecedente como valores durante o semestre; só considerarei promover um deles a classe numa reformulação pontual se isso ajudar a demonstrar um conceito do tópico. O âmbito funcional fica no essencial: selecionar + pré-visualizar impacto + confirmar + guardar ficha (em texto), sem exportação real de PDF, sem persistência, sem UI avançada e sem equilíbrio "realista" de estatísticas da personagem. Priorizarei estudar e aplicar, ao longo dos próximos tópicos, as matérias deleis, aceitando que melhorias de validação e funcionalidades só serão implementadas caso se revelem simples para exemplificar a matéria que estiver a estudar.

Evidências incluídas no documento: estou a incluir storyboards e mockups, excertos de código (menos de 60 linhas).

Princípios utilizados explicitamente nesta reflexão, acima: **classe vs. instância**; **abstração**; **responsabilidade única**. Identifiquei-os com **índice único**.

O mapa transforma uma observação vaga numa decisão técnica. A síntese liga essa decisão ao diagrama, ao código e aos restantes artefactos.

Um GPT genérico também pode debater estas matérias com os estudantes, não teria sido preciso eu fornecer o Sócrates Digital e o Aristóteles Estoico. Contudo, se o estudante não lhe desse o contexto da disciplina e da tarefa, não obteria uma interação adequada. Num grupo de muitos estudantes, a disparidade de qualidade nessa interação seria enorme. Os GPT preparados reduzem essa barreira. Contudo, há uma questão institucional e de proteção de dados. Não é aceitável, à luz do regulamento geral de proteção de dados (RGPD, o bem-intencionado mas malfadado) que para usar um GPT específico cada estudante tenha de criar uma conta e enviar dados pessoais para um serviço fora da União Europeia. Além disso, se pudermos evitar

despesas aos estudantes, será adequado dar-lhes essa liberdade. Assim, não é obrigatório usar os GPT que criei na OpenAI como *custom GPTs*. Num Grimório de Interpelações existente na unidade curricular, publico as interpelações (os *prompts*) de todos os GPT que uso. E também os ficheiros de referência, se necessário, permitindo aos estudantes reproduzir a interação noutro sistema. Assim, podem recorrer em alternativa à OpenAI a um serviço alojado na União Europeia, em Portugal, na própria instituição, noutro fornecedor escolhido pelo estudante ou mesmo num modelo de linguagem executado no seu computador. Ferramentas como o Ollama e o LM Studio já permitem executar modelos localmente, sem enviar as conversas para um serviço externo.

O *feedback* automático para todos muda o trabalho do professor

O ganho logístico de assumir e encorajar o uso dos GPT é imediato: cada estudante pode obter *feedback* detalhado, personalizado, específico, sem aguardar que um docente encontre tempo para chegar ao seu projeto e o analisar em toda a profundidade. Mas a presença dos GPT na unidade curricular cria uma exigência permanente para professores e tutores... e alunos. Temos todos de demonstrar o nosso valor para além do que o GPT já proporciona.

Isto tem consequências interessantes nas dinâmicas em fórum. Perante uma dúvida, posso responder: «Isto é o que o Sócrates Digital diz quando lhe apresento a sua questão». E depois acrescentar: «E isto é o que tenho a dizer da minha parte». A primeira resposta resolve habitualmente aspetos formais ou expectativas genéricas ou canónicas: uma omissão, um critério, um exemplo. A segunda pode reconhecer uma intenção particular, relacioná-la com algo que aconteceu noutro projeto, distinguir uma dificuldade conceptual de uma decisão razoável, perceber que o estudante está a evitar um risco importante ou dizer-lhe, com base em experiência, onde vale a pena insistir.

Este uso aberto da IA encoraja os estudantes a compartilharem as suas consultas, dúvidas e resultados. Pois se «até o professor o faz...» a conversa deixa de ficar escondida por receio de parecer ilegítima. Não, colegas, não basta dizer aos estudantes que podem usar, porque os alunos têm anos de constatar a duras penas que nem tudo o que os professores dizem é o que cumprem. É importante além de o dizer, fazê-lo. Os colegas também podem intervir da mesma maneira. Podem comparar formas de interpelar, contestar uma resposta ou perceber que enfrentam problemas semelhantes. E o estudante não precisa de permanecer em silêncio à espera da validação do professor para cada pequeno passo.

O trabalho docente não desaparece! Fica é mais exposto. Se a minha intervenção se limitar a fazer o mesmo que o GPT ou pior, a inutilidade disso torna-se evidente. A riqueza da vivência e experiência humana tem de aparecer no que se escolhe, no que se valoriza, na leitura da situação e na forma como se ajuda aquela pessoa e aquela turma a avançar.

O que correu bem

O primeiro resultado foi constatar a viabilidade. Um processo que anteriormente nem se podia pedir, porque exigiria várias semanas e distribuiria o acompanhamento de forma muito desigual coube no esforço previsto para duas semanas. Os estudantes tinham acesso a *feedback* imediato e mais detalhado do que aquele que a equipa docente poderia proporcionar individualmente a mais de duzentas pessoas.

Os temas propostos foram ricos e diversos. A exigência de relevância pessoal ou profissional deu aos projetos uma direção que uma lista comum de temas não teria... e nos que não deu, permitiu cedo (ao fim dessas duas semanas) alertar os estudantes para esse problema. As várias concretizações permitiram que essa direção fosse discutida, em

vez de permanecer escondida, subentendida numa declaração inicial que ninguém voltaria a examinar.

Também se tornou visível na turma a diversidade de perspectivas e a clonagem de ideias. Projetos diferentes fazem tropeçar os mesmos princípios em lugares diferentes; projetos idênticos com progressões idênticas destapam a falta de autenticidade. Conceitos abstratos da matéria, como a “responsabilidade única”, parecem evidentes num exemplo preparado para uma aula, mas num projeto pessoal, aquilo que conta como «uma responsabilidade» torna-se discutível e precisa de ser defendido.

Estes exemplos são meros indícios operacionais, observações que fiz. Ainda não demonstram que os estudantes aprenderam mais ou melhor. Demonstram que foi possível proporcionar à generalidade da turma um tipo de percurso e de *feedback* que, nesta escala, antes estaria reservado a alguns ou nem se ousaria tentar proporcionar.

O que correu mal e por isso mesmo foi útil

Acima, no que correu bem, já incluí problemas: estudantes que propunham projetos sem relevância pessoal, ideias clonadas, como lhes chamei, por terem sido pedidas ao GPT sem introspeção ou copiadas de colegas. Isso é correr bem! Porque um processo que revela problemas cedo, não tarde, é um processo bom, que permite introduzir correções atempadas. O que correu mal são problemas não de indivíduos, mas do próprio processo. São os aspetos que não permitem que emergja o valor pretendido.

O que mais me custou foi o ter ingenuamente incluído um pedido de síntese final. No pedido dela, deixei deliberadamente um espaço de voz menos estruturado. Queria que os estudantes explicassem, fora do espartilho de uma tabela, a incoerência encontrada, as consequências, o limite assumido e a relação com os princípios da matéria. Às vezes foi isso que fizeram, mas em muitas entregas aconteceu outra coisa: texto

genérico, cheio de vocabulário correto, provavelmente produzido por um GPT sem apropriação pessoal. «Otimização», «reflexão crítica», «abordagem inovadora», «aplicação eficiente dos princípios». As palavras estavam lá; o projeto pessoal, a perspectiva do indivíduo desaparecera.

OK, dirão: mas então aqui também é bom que isto tenha sido revelado cedo! Não: num projeto genérico, o estudante pode achar que o cunho pessoal não é importante e qualquer tema serve para aprender. Há uma margem de ingenuidade de que possa estar a agir bem (não estando), mas aqui não fica nenhum estudante em dúvida de que está a fugir à reflexão.

O que correu mal foi que não planeei o meu ato de avaliação a contar com isso. Devia tê-lo feito. Só o fiz a meio do processo, quando me imbuí de consciência do que se passava. Se eu avaliar esse texto como avaliaria uma reflexão presumivelmente escrita sem IA, entro numa espiral de frustração. Tenho de refletir sobre se por trás de uma generalidade se esconde uma intenção ou apenas vacuidade. Tenho de pensar se aquele “trade-off” veio do GPT acefalamente ou se o aluno gostou da ideia e não se incomodou com o termo. Em geral, tenho de andar a produzir *feedback* para um esforço que não existiu. Então, neste casos mais frágeis à retórica vazia, o foco da avaliação tem de ir diretamente à jugular: há sentido pessoal, há um propósito pertencente claramente ao proponente, há decisão, há aspetos específicos do projeto ligados ao que se viveu a fazê-lo? Há agência? O *feedback* só avança se isso for evidente, senão é isso que se discute, nada mais.

Quando a reflexão fica genérica

RECONSTRUÇÃO SINTÉTICA — NÃO É UM DIÁLOGO REAL

E

Reflexão entregue

«A utilização da inteligência artificial permitiu otimizar o processo de desenvolvimento, promover uma reflexão crítica e aplicar os princípios de POO de forma eficiente e inovadora.»

D

Feedback docente

Que decisão tua mudou?
Onde aparece no diagrama ou no código?
Que consequência identificaste?
Que limite aceitaste?
Sem isso, esta reflexão ainda não fala do teu projeto.

O momento útil é a intervenção que exige ligação entre palavras, decisões e artefactos.

Recriação sintética: não reproduz qualquer diálogo ou trabalho efetivo de um estudante.

Uma formulação vaga ou feita de palavras da moda deve receber uma crítica inequívoca. Esse *feedback* inicial marca o tom para o resto do semestre. Mostra que os critérios não eram decoração para a rubrica e esclarece o que entendo por reflexão: estabelecer relações que só aquela pessoa pode estabelecer entre a sua intenção, as suas escolhas e aquilo que construiu.

O mesmo ocorreu, em menor grau mas ainda assim demasiado frequente nas tabelas de incoerências, consequências, limitações e adoção de princípios da matéria. Requereu igualmente uma reorganização da abordagem de avaliação e *feedback* para se focar nestas qualidades como ponto prévio, antes de se desgastar a discutir vacuidades.

Também aqui há trabalho de reflexão quanto ao desenho instrucional. Talvez seja possível minorar estas atitudes com contraexemplos, não apenas exemplos de execução. Talvez seja possível aprimorar o *feedback* por GPT para alertar previamente para este tipo de defeitos,

não apenas no momento irreversível da avaliação humana. Talvez haja outras abordagens viáveis.

As ideias bonitas só são a sério quando escrevemos o enunciado

O caso de Programação por Objetos não oferece uma receita para todas as áreas. Mas espero que possa ser útil para muitas! Mostra uma abordagem que pode ser adaptada: partir de algo que importe ao estudante; obrigar a ideia a atravessar concretizações diferentes, transformações cognitivas variadas; usar a IA para proporcionar interlocução frequente; procurar incoerências entre intenção e realização; analisar consequências dessas incoerências; decidir quais os limites aceites para a resolução dessas incoerências; reservar a intervenção docente para o sentido crítico, o propósito, a individualização e a relação entre aquela pessoa e o que está a construir.

Foi assim que uma tarefa aparentemente simples — propor um tema de projeto — passou a ocupar duas semanas sem se transformar em enchimento de chouriços curricular. A proposta de um tema de projeto tornou-se parte da própria aprendizagem. O estudante começou a encontrar a programação por objetos dentro do problema que quis construir, antes de o projeto ter dimensão suficiente para esconder as decisões sob centenas de linhas de código.

No texto anterior, terminei perguntando que processo leva o estudante a valorar, decidir, rever e assumir. Esta foi uma resposta possível, aplicada em condições reais e ainda imperfeita.

É também aqui que as tangas pedagógicas se convertem na roupagem da realidade. Depois de defendermos compreensão, apropriação e sentido crítico, temos de decidir o que acontece na primeira semana, o que se entrega na segunda, que *feedback* chega a todos, onde entra o

professor e o que fazemos quando a prosa impecável não contém alma lá dentro.

Referências e ligações

- Papert, S., & Harel, I. (1991). *Situating Constructionism*. Primeiro capítulo de *Constructionism*.
- Wilensky, U. (1991). *Abstract Meditations on the Concrete and Concrete Implications for Mathematics Education*. Em I. Harel & S. Papert (Eds.), *Constructionism*, pp. 193–204.
- European Data Protection Board. [International transfers and international cooperation](#).
- [Ollama](#) e [LM Studio](#), opções para executar modelos localmente.

Young Builders is a reader-supported publication. To receive new posts and support my work, consider becoming a free or paid subscriber.

✓ **Subscribed**