A Geração Aumentada por Recuperação (RAG) é algo que tem se mostrado bem útil para melhorar as respostas dos modelos de linguagem. A ideia é simples, mas muito boa: antes de responder, o modelo dá uma olhada em documentos relevantes, o que ajuda a evitar aquelas “alucinações” em que a IA inventa dados do nada. Um exemplo interessante é o projeto Jugalbandi AI, que usa essa técnica. Só que, para isso funcionar, os documentos são divididos em partes menores, o que facilita o processamento, mas também pode deixar o projeto mais complexo.

No caso do Backstage da Spotify, eles automatizaram a geração de descritores de entidades, o que facilita muito a manutenção dos catálogos de ativos. Isso ajuda a evitar erros e deixa a atualização dos dados mais simples. Eles usam metadados e tags que já existem, então ninguém precisa ficar reinventando a roda. Parece uma solução bem prática, mas se os dados estiverem bagunçados, pode dar problema.

Agora, misturar técnicas tradicionais de Processamento de Linguagem Natural (PLN) com os LLMs é uma combinação interessante. Os LLMs são muito poderosos, mas, para algumas tarefas mais simples, como sumarização de texto ou análise de tendências, as técnicas tradicionais são mais baratas e eficientes. O truque é saber quando usar uma ou outra.

Outro ponto interessante é a tal da conformidade contínua. É basicamente integrar segurança e padrões de regulamentação no software desde o começo do desenvolvimento. Isso pode evitar muita dor de cabeça no futuro, principalmente para quem trabalha com softwares críticos. Não é fácil, mas acho que vale muito a pena.

As funções de edge, por outro lado, são bem legais porque permitem que o código rode mais perto do usuário, diminuindo a latência. Isso é ótimo para quem precisa de respostas rápidas, mas elas têm algumas limitações, como funcionar melhor em processos sem estado e terem capacidade computacional mais baixa. Não são a solução para tudo, mas quando encaixam, fazem a diferença.

A ideia de ter “campeões” de segurança nas equipes também é muito boa. Essas pessoas ficam responsáveis por garantir que as boas práticas de segurança sejam seguidas. Isso ajuda a criar uma cultura de segurança, que muitas vezes é deixada de lado por conta de prazos apertados. Mas, claro, é uma baita responsabilidade, e a equipe precisa realmente valorizar essa pessoa para que as coisas funcionem bem.

Converter consultas de linguagem natural para SQL é uma daquelas coisas que parecem simples, mas faz uma baita diferença. O framework Vanna é um exemplo disso, ajudando quem não manja de SQL a criar consultas. Ele usa RAG para melhorar a precisão dessas consultas, o que deve economizar um bom tempo no desenvolvimento.

Outra ideia que eu acho interessante é focar na saúde do sistema, em vez de ficar obcecado em pagar a “dívida técnica”. A ideia é manter o software saudável e funcionando bem, porque, no fim das contas, isso é mais importante do que simplesmente eliminar dívida técnica, que nem sempre reflete o real estado do sistema.

Os assistentes de IA já provaram ser muito úteis para programadores, mas agora a ideia é expandir isso pra ajudar equipes inteiras. Assistentes como o GitHub Copilot já aumentam bastante a produtividade individual, e faz sentido pensar em como escalar isso pro time todo. Depende de como cada equipe implementa, mas é um caminho que parece natural.

Um conceito que me surpreendeu foi usar grafos para analisar conversas com chatbots. Isso ajuda a identificar padrões e melhorar o produto com base nos dados de interação. Deve ser meio complicado de implementar, mas acho que pode dar uma visão muito interessante de como os usuários estão interagindo com o sistema.

O CloudEvents é outra coisa que está ganhando bastante força, porque ajuda a padronizar dados de eventos entre diferentes plataformas de nuvem. Isso facilita a comunicação entre sistemas e é cada vez mais necessário, já que os serviços estão mais distribuídos. Como já foi adotado por muitos provedores, facilita ainda mais a integração.

A computação baseada em Arm também está crescendo rápido, especialmente por ser mais eficiente em termos de energia e custo. Grandes provedores como AWS, Azure e GCP já oferecem instâncias Arm, e muita gente está migrando para elas, principalmente serviços que usam JVM ou bancos de dados. Acho que o Arm tem muito potencial, mas ainda depende de como as aplicações vão se adaptar.

Os Aplicativos de Contêiner do Azure são uma alternativa mais simples ao Azure Kubernetes Service (AKS), que pode ser bem complicado de gerenciar. Eles facilitam a implantação de contêineres, mas têm algumas limitações em termos de flexibilidade. Para quem busca simplicidade, parece uma boa opção.

O Serviço de Open AI do Azure é legal porque traz os modelos da OpenAI, como o GPT-4 e o DALL-E, pro ambiente do Azure. E como o Azure já tem uma série de medidas de segurança e conformidade, isso faz com que seja uma boa escolha para empresas que já usam a plataforma.

Por fim, o DataHub tem sido muito útil pra gente, facilitando o gerenciamento e a governança de dados. As melhorias recentes, como a personalização de metadados e a arquitetura baseada em plugins, deixaram a plataforma ainda mais flexível. Tem sido essencial para garantir que os dados estejam sempre bem organizados e acessíveis.

Esses conceitos e ferramentas mostram como a tecnologia está evoluindo rápido, automatizando processos e ajudando as equipes a focarem mais no que realmente importa, sem precisar se preocupar tanto com tarefas repetitivas ou complexas demais.