Agentes de Inteligência Artificial para Pesquisa

Seminário Discente DCP-USP 2026 Minicurso de Curta Duração

Manoel Galdino
Professor no DCP-USP

Felipe Lamarca
Mestrando no IESP-UERJ

Tales Mançano
Mestrando no DCP-USP

2026-08-24

1. Visão Geral e Justificativa

No último ano, o uso mais interessante dos modelos de linguagem (LLMs) transicionou dos chatbots para agentes: os modelos ganharam a capacidade de executar comandos no computador, consultar arquivos e encontrar informações autonomamente. Na pesquisa acadêmica, isso significa executar planos com várias etapas, manipular dados empíricos, usar ferramentas externas (editar e ler arquivos, chamar APIs, pesquisar na internet) e testar os resultados das próprias tarefas. Isso já é útil para diversas tarefas que antes tomavam muito tempo e agora saem com custo bem mais baixo: análise exploratória de dados, transcrição de entrevistas, organização e triagem de arquivos extensos, classificação de notícias e outros documentos, e criação de bases de dados, entre outras.

Este minicurso, oferecido no âmbito do **Seminário Discente DCP-USP 2026**, tem como propósito introduzir pesquisadores aos fundamentos para usar agentes de IA de forma rigorosa — evitando problemas comuns como alucinações, falta de registro do que foi feito e falta de replicabilidade, e garantindo a possibilidade de revisar o trabalho depois. Isso passa por *code as policy*: regras rígidas sobre o que um agente pode e não pode fazer, para evitar problemas de reprodutibilidade e rigor na pesquisa com agentes.

---

2. Objetivos de Aprendizagem

Ao final do workshop, os participantes deverão ser capazes de:

1. **Reconhecer** tarefas de pesquisa em que a automação com agentes é vantajosa — transcrição, classificação de documentos, organização de arquivos extensos, análise exploratória de dados.
2. **Configurar** o *harness* de um agente — permissões, isolamento e containerização — para rodar com segurança.
3. **Identificar** quando confiar no julgamento do modelo (instrução em prosa) e quando travar o comportamento com uma regra fixa (*hook*, *code as policy*).
4. **Escrever** hooks determinísticos, *skills* reutilizáveis e um `AGENTS.md` que funcione de verdade — evitando as armadilhas mais comuns.
5. **Avaliar** e integrar ao fluxo de pesquisa uma ferramenta de IA já existente.

---

3. Informações Operacionais

- **Carga Horária**: 3 horas (2 sessões de 1h30).
- **Público-Alvo**: Alunos de pós-graduação, pesquisadores e docentes em Ciência Política, Relações Internacionais, Sociologia e áreas afins.
- **Pré-Requisitos**: Conhecimentos básicos de uso de computador e manipulação de arquivos. Não há exigência de experiência prévia em programação, embora familiaridade com R, Python ou terminal de comando seja benéfica.
- **Formato**: Aulas expositivas e demonstrações ao vivo.

---

4. Programa do Curso

Sessão 1: Introdução aos Agentes (1h30)

Foco: caráter estritamente conceitual — o que muda quando um LLM ganha capacidade de agir. Nenhuma habilidade prática é ensinada nesta sessão.

1. **LLMs como Agentes**
   - LLMs como agentes: o que muda quando o modelo tem acesso a terminal, edita arquivos locais ou em rede, e executa tarefas de várias etapas sem supervisão a cada passo.
2. **Panorama de Modelos e Ferramentas**
   - Principais famílias de modelos hoje (Claude, GPT, Gemini, DeepSeek, modelos abertos) e o que muda entre elas na prática de pesquisa.
   - Agentes de terminal e de editor (Open Code, Claude Code, entre outros): o que são e como se diferenciam de um chatbot comum.
3. **Benefícios, Vantagens e Custos**
   - Que tarefas de pesquisa se beneficiam de automação com agentes — e quais não valem o esforço de configurar um agente.
   - Modelos de cobrança: assinatura de aplicativo vs. cobrança por uso de API (tokens); ordem de grandeza de custo para tarefas típicas de pesquisa.
4. **O Harness do Agente: Isolamento e Containerização**
   - *Harness*: o ambiente que envolve o modelo e viabiliza sua execução — quem controla permissões de leitura/escrita, o que o agente pode acessar, quem decide quando parar.
   - Containerização: rodar agentes em sandbox/container limita o raio de dano de uma ação indevida.
5. **Instrução em Prosa vs. Restrições Determinísticas**
   - Quando confiar no julgamento do modelo (instrução em prosa) e quando impor uma regra fixa que executa sempre da mesma forma.
   - Por que a pesquisa empírica exige, em certos pontos do processo, uma previsibilidade que demanda regras determinísticas.
   - Gancho para a Sessão 2: essa distinção é a base de *hooks* e *code as policy*.

Sessão 2: Habilidades Práticas e Governança (1h30)

Foco: colocar a mão na massa nas ferramentas que tornam um agente confiável e útil para a pesquisa.

1. **Skills: Instruções Reutilizáveis**
   - O que são *skills*: instruções empacotadas e reutilizáveis para tarefas de pesquisa recorrentes.
   - Para quem quiser ir além: orquestração de múltiplos agentes trabalhando em conjunto numa mesma tarefa — por exemplo, separar um agente que implementa de outro que só valida o que foi feito, mantendo o contexto de cada etapa enxuto e a revisão independente da execução.
2. **Controle Determinístico com Hooks**
   - *Hooks* de agente: validações e bloqueios (*guardrails*) que sempre disparam antes ou depois da execução do modelo, independente do "bom senso" do modelo — a mesma lógica dos *git hooks*, usada aqui apenas como analogia conceitual para intuir o mecanismo, sem exigir conhecimento prévio de Git.
   - *Code as policy*: checagem de integridade de dados, validação de formato e auditoria de commits como regras executáveis, não como pedido em prosa — apresentado via demonstração ao vivo pelos ministrantes, não como exercício prático dos participantes.
3. **MCP, Linha de Comando e Segurança**
   - Model Context Protocol (MCP): como conectar agentes a bases de dados locais, APIs (inclusive governamentais) e ferramentas de busca de forma padronizada.
   - Boas práticas de segurança: isolamento de chaves de API e permissões de execução — por que "menos acesso" é o padrão seguro.
4. **`AGENTS.md` e Reprodutibilidade**
   - Estrutura, nível de detalhe e armadilhas comuns ao instruir um agente sobre como operar em um repositório ou projeto de pesquisa.
   - Rastreabilidade: logs de execução e padrões de versionamento que permitem revisar o trabalho do agente depois.
5. **Software com IA para Pesquisa — Aplicações Exemplares**
   - O que caracteriza uma boa aplicação de IA no fluxo de pesquisa: respostas rastreáveis até a fonte, integração com as ferramentas que o pesquisador já usa, e transparência sobre o que foi automatizado.
   - Categorias com boas alternativas de IA hoje: gestão de referências (Zotero + Beaver), transcrição de áudio, análise de dados, etc.
   - Gerenciamento de contexto como boa prática recorrente nessas aplicações: por que separar o agente que executa do agente que audita o resultado reduz viés de confirmação e mantém cada etapa com um contexto mais enxuto e específico.