

ProfAI – Agente codocente para mediação pedagógica em ambientes virtuais de aprendizagem

Gabriel Aragão¹, Leonel Morgado²

¹ Universidade Aberta, Lisboa, Portugal
gabriel.aragao.comp@gmail.com

² LE@D, CIAC, CEG, Universidade Aberta, Coimbra, Portugal & INESC TEC, Porto, Portugal
leonel.morgado@uab.pt

Resumo

Os sistemas de gestão da aprendizagem (LMS), como o Moodle, sistematizam tarefas, mas pouco oferecem para sustentar dinamização individualizada de grandes números de estudantes pelos docentes. Este artigo apresenta o ProfAI, um agente inteligente concebido para apoiar atividades pedagógicas no Moodle. O sistema combina *Retrieval-Augmented Generation* (RAG) para conteúdos estáticos e integração com o LMS para interações dinâmicas. A execução ocorre em infraestrutura local ou organizacional, sem necessidade de recurso a serviços online exteriores às instituições, para conformidade com os regulamentos de proteção de dados pessoais. O ProfAI assegura aos estudantes um *feedback* individualizado e atua como assistente docente na dinâmica pedagógica. Pretende-se assim contribuir para modelos de emprego de agentes inteligentes interativos no e-learning, de forma segura e replicável.

palavras-chave: agente, Moodle, LLM, RAG, AI, e-learning.

Title: ProfAI - A practical approach to automating tasks in the virtual learning environment using agents

Abstract: Learning Management Systems (LMS), such as Moodle, systematize tasks but offer limited support for individualized student large-scale dynamics by instructors. This paper presents ProfAI, an intelligent agent designed to support pedagogical activities within Moodle. The system combines Retrieval-Augmented Generation (RAG) for static content with LMS integration for dynamic interactions. Execution takes place on local or institutional infrastructure, avoiding reliance on external online services and ensuring compliance with personal data protection regulations. ProfAI provides students with individualized feedback and acts as a teaching assistant within the pedagogical dynamic, contributing to secure and replicable models of intelligent interactive agents in e-learning.

keywords: agent, Moodle, LLM, RAG, AI, e-learning

1. Introdução

Nas últimas duas décadas, o ensino online consolidou-se como uma componente essencial das práticas educativas em instituições de ensino superior. Ambientes virtuais de aprendizagem, como o Moodle, tornaram-se plataformas centrais para a organização de conteúdos, comunicação entre docentes e estudantes e acompanhamento de atividades avaliativas. Essas plataformas ampliaram o acesso à educação e permitiram a gestão digital de cursos em larga escala. Contudo, apesar dos avanços tecnológicos, a mediação pedagógica em ambientes virtuais continua a depender fortemente da intervenção humana, especialmente na orquestração das interações e na produção de *feedback* individualizado.

Diversos estudos evidenciam que os docentes enfrentam um esforço cognitivo e temporal significativo ao tentar acompanhar as necessidades específicas de cada estudante, configurando um problema social de relevância crescente [Holstein et al., 2019; Prieto et al., 2019]. Os docentes expressam a necessidade de "clonar-se temporariamente" para conseguir fornecer apoio individualizado a múltiplos estudantes simultaneamente, reconhecendo que as suas limitações de tempo durante sessões de laboratório dificultam a monitorização das atividades dos estudantes e a provisão de *feedback* atempado [Holstein et al., 2019]. A maioria das ferramentas automatizadas disponíveis, como *chatbots* educacionais e sistemas de *learning analytics*, exige configuração e manutenção manuais, baseando-se em fluxos rígidos de regras e respostas pré-programadas [Maier & Klotz, 2022; Kuhail et al., 2022]. Tais soluções apresentam forte dependência de interação contínua iniciada pelo utilizador [Kuhail et al., 2022], limitando a sua eficácia em contextos que exigem acompanhamento pedagógico prolongado e proativo.

Diante desse cenário, torna-se necessário repensar o papel dos agentes inteligentes na educação, evoluindo de ferramentas reativas, que apenas respondem a comandos, para sistemas proativos de mediação pedagógica. Propõe-se, assim, o conceito de agente codocente: uma entidade de software dotada de capacidades cognitivas e operacionais capazes de complementar o trabalho docente, compreendendo o contexto do curso e agindo com autonomia controlada no ambiente virtual de aprendizagem. Diferentemente dos *chatbots* convencionais que operam de forma reativa e dependem de interação contínua iniciada pelo utilizador [Kuhail et al., 2022], o agente codocente atua proactivamente: monitoriza o estado da aprendizagem, identifica necessidades emergentes sem solicitação explícita e toma iniciativa pedagógica de forma autónoma. Dessa forma, o agente codocente reduz substancialmente a carga de orquestração manual do docente, permitindo escalabilidade do apoio individualizado sem comprometer a qualidade pedagógica que caracteriza a mediação docente efetiva.

Este artigo tem como objetivo principal explorar a viabilização técnica de um agente desse tipo, denominado ProfAI, e sua prototipagem, capaz de mitigar as limitações identificadas na prática docente online. O sistema combina arquitetura RAG com integração dinâmica com o Moodle, execução local (nos equipamentos da instituição de ensino superior ou do docente) e containerização via Docker e Ollama, garantindo conformidade com o Regulamento Geral de Proteção de Dados (RGPD). Ao propor o ProfAI, busca-se demonstrar como um agente de IA pode atuar como codocente digital,

providenciando um apoio personalizado aos estudantes, ainda que potencialmente em grande número.

2. Cenário atual

Nos últimos anos, tem-se observado um crescimento expressivo do uso de inteligência artificial (IA) em ambientes educacionais, sobretudo com o advento de agentes inteligentes e grandes modelos de linguagem (LLM, *large language models*) [Labadze, 2023]. No entanto, nem sempre o uso de agentes em educação esteve associado a IA generativa. Durante a última década, a tecnologia de *chatbots* foi amplamente utilizada como meio de interação automatizada em sistemas de e-learning.

De acordo com Adamopoulou & Moussiades [2020], os chatbots, também designados *smart bots*, *interactive bots* ou *digital assistants*, são programas computacionais concebidos para simular interações humanas por meio da linguagem natural, respondendo a utilizadores com base em dados e regras pré-definidas. Esses sistemas funcionam como intermediários entre o utilizador e uma base de conhecimento, podendo aceder diretamente a informações ou ferramentas relevantes para formular respostas. Após sua consolidação em contextos empresariais e de suporte técnico, vêm sendo progressivamente incorporados em ambientes de aprendizagem virtual para tarefas administrativas, como resposta a perguntas frequentes e apoio técnico a estudantes.

Contudo, conforme analisado por Winkler & Söllner [2018], a maioria desses *chatbots* era baseada em regras, apresentando forte dependência de fluxos de diálogo previamente configurados. Essa característica limitava a sua eficácia em interações complexas ou não previstas, comprometendo a continuidade e a naturalidade da conversa com o utilizador.

Em contraste, os agentes inteligentes representam uma evolução conceitual e técnica. São sistemas capazes de perceber o ambiente, processar informações e agir de forma orientada a objetivos, buscando atingir resultados específicos [Russell & Norvig, 2021]. No contexto digital, tais agentes operam como entidades de software que interagem com utilizadores e sistemas, interpretando dados contextuais para oferecer respostas ou executar ações de forma autónoma. Um marco fundamental para o avanço desses agentes foi a introdução da arquitetura Transformer, apresentada por Vaswani et al. [2017] no artigo “Attention Is All You Need”. Essa arquitetura revolucionou o processamento de linguagem natural ao substituir estruturas recorrentes por mecanismos de atenção, permitindo a captura de dependências contextuais em grandes volumes de texto e o treino paralelo de modelos. Com base nesse avanço, a OpenAI lançou, em 2018, o GPT-1 (Generative Pre-trained Transformer), o primeiro modelo de larga escala a combinar pré-treino não supervisionado e ajuste fino em tarefas linguísticas [Radford et al., 2018]. A consolidação desses desenvolvimentos culminou no lançamento do ChatGPT, em 2022, uma interface conversacional para um modelo GPT (na altura, o GPT-3), o primeiro caso de sucesso global de interação conversacional com os LLM. Desde então, observou-se uma rápida disseminação do uso desses modelos em múltiplos domínios, inclusive na educação.

Apesar dos avanços dos *bots* ainda restam desafios. Um dos grandes desafios é o correto tratamento dos dados pessoais, visto que esses sistemas interagem com estudantes e recolhem dados durante as conversações [Labadze et al., 2023]. Quando sistemas educacionais transmitem dados identificáveis a pessoas (incluindo nomes, avaliações e interações) para serviços externos de processamento de LLM, há risco significativo de violação do RGPD, como ilustrado pelo bloqueio temporário do ChatGPT em Itália por não conformidade com as regras de proteção de dados. [Labadze et al., 2023].

E relativamente aos *bots* sem uso de inteligência artificial generativa, há uma grande dependência na configuração de fluxos e uma limitação do conteúdo a que o *bot* consegue responder [Thorat & Jadhav, 2020]. Uma outra questão é a dependência de interação contínua entre o utilizador e o *bot*, necessária para que o sistema execute ações ou mantenha o contexto conversacional [Kuhail et al., 2022]. Esta dependência de interação reativa contrasta com as necessidades de ambientes educacionais que exigem acompanhamento pedagógico prolongado, atempado e proativo, onde a exigência constante de iniciativa do utilizador limita a eficácia do sistema.

3. Problema, conceito e proposta

3.1. Problema e conceito

O uso de *bots* “clássicos” em e-learning tende a exigir configuração e manutenção manuais extensas (por exemplo, definição de fluxos e regras de decisão), o que limita a sua capacidade de generalização e de adaptação a situações novas. As revisões recentes sobre *feedback* personalizado em ambientes digitais mostram um predomínio de soluções baseadas em regras ($\approx 74\%$), frequentemente implementadas como programas autónomos que consomem dados do LMS e disparam e-mails semanais com regras “se-então” (e.g., OnTask, SARA), precisamente o tipo de configuração que demanda trabalho manual do docente ou da equipa técnica [Maier & Klotz, 2022]. Assim como o estudo citado anteriormente, os factos são corroborados de maneira que a maioria dos *chatbots* (88%) segue fluxos pré-definidos baseados em regras estritas de condicionais lógicas, mostrando uma vasta quantidade de *chatbots* que ainda seguem esse modelo [Kuhail et al., 2022].

Além disso, a geração de *feedback* personalizado raramente é automatizada de forma contextual: as revisões sistemáticas demonstram que a maioria das abordagens de *feedback* ainda dependem de ajustes manuais e da interpretação humana, limitando a escalabilidade das práticas pedagógicas [Maier & Klotz, 2022].

Assim, a natureza manual desses processos constitui um dos principais entraves à personalização e à sustentabilidade do ensino mediado por tecnologia, reforçando a necessidade de agentes inteligentes capazes de atuar como codocentes no apoio ao professor. Adicionalmente, revisões sistemáticas de *chatbots* educacionais revelam limitações significativas mesmo em sistemas que utilizam *machine learning* para otimizar respostas: a limitação mais recorrente é o treino com *datasets* insuficientes ou inadequados, o que resulta em incapacidade de responder adequadamente às questões dos

estudantes, gerando frustração e comprometimento do processo de aprendizagem [Kuhail et al., 2022]. Também se notou que a comunicação com um *chatbot* fez perder o interesse nas atividades em questão, o que com um humano fez com que não houvesse essa perda de interesse, mostrando que esse estilo conversacional via *chatbot* pode trazer impacto negativo no aprendizado [Kuhail et al., 2022].

Esses fatores mostram que uma possível automação pode permitir uma mediação inteligente que atue de maneira proativa, sem a necessidade de uma conversa explícita, gerando uma possível solução aos problemas citados no parágrafo anterior. É também necessária uma ferramenta que possua contexto da disciplina como um todo, o seu conteúdo, o cronograma, os alunos, o histórico dos alunos, etc. Outro fator importante é a redução da configuração manual feita por um docente, é necessário que um sistema consiga se autorregular conforme necessário, ou seja, um sistema que consegue identificar quando e como agir de acordo com as necessidades do ambiente virtual (como uma pergunta de um aluno que necessita resposta) e do momento temporal (como uma data para se iniciar avaliações).

Com isso surge a proposta de um codocente ou coprofessor, que possui o contexto necessário para agir e compreende quando e como agir no ambiente virtual de aprendizagem.

Para se chegar nesse agente que se autorregula é preciso que ele não tenha vieses e consiga trabalhar com diversos temas. De acordo com Shulman [1986] um bom professor precisa ter alguns conhecimentos: Conhecimento do conteúdo (*Subject Matter Knowledge*), Conhecimento Pedagógico do Conteúdo (*Pedagogical Content Knowledge – PCK*), Conhecimento Curricular (*Curricular Knowledge*) e de extrema importância, o professor deve saber transformar o conhecimento em ensino compreensível.

Para se chegar nesse “bom” professor é necessária uma vasta quantidade de dados, à qual um professor humano foi exposto durante muitos anos. Graças à tecnologia dos LLM é possível ter acesso a esses dados de uma maneira facilmente acessível a uma linguagem natural. Ainda não existem estudos que permitam concluir que os LLM tenham o mesmo nível de conhecimento que um professor de ensino superior, mas existem estudos que relatam bons resultados dos LLM comparativamente com profissionais doutorados, embora vários sejam internos às empresas como a OpenAI [Learning to reason with LLMs, 2024].

Além do conhecimento necessário para ser um docente, o codecente proposto também precisa de agir por sua iniciativa e não apenas reagir. Assim como um professor tem capacidades físicas de se comunicar e tomar ações, como por exemplo corrigir provas ou publicar conteúdos no ambiente virtual de aprendizagem, assim também o codocente precisa de ter a capacidade de agir: ter agência. Podemos entender o agente codocente como tendo duas partes principais, o cérebro e os braços/pernas. O cérebro é a LLM, que contém o conhecimento necessário para “pensar” como um professor. Os braços/pernas são as ferramentas com as quais o agente tem acesso para agir, como por exemplo acesso e permissões ao ambiente virtual de aprendizagem para poder publicar uma mensagem

no fórum. Essas ferramentas devem permitir com que o agente interaja com o ambiente e com os alunos: o agente deve conseguir entender o que se passa nas disciplinas, assim como conseguir dar resposta, seja a mensagens de alunos, seja à correção de um trabalho ou à publicação de materiais.

3.2. Proposta (ProfAI)

Para responder aos desafios de orquestração manual e falta de personalização identificados na literatura apresentada, o ProfAI foi concebido como um agente codocente capaz de atuar proativamente em ambientes virtuais de aprendizagem. O sistema combina uma arquitetura RAG (Retrieval-Augmented Generation) — uma técnica que permite ao modelo de linguagem obter e utilizar conteúdo específico, por exemplo materiais de estudo fornecidos pelo docente como fontes primárias de conhecimento, em vez de depender exclusivamente do seu treino geral [Thoeni & Fryer, 2025] — com integração dinâmica ao Moodle LMS, operando de forma segura e replicável em infraestrutura local.

O projeto foi desenvolvido em Node.js (Typescript) com uso de frameworks para rápida prototipagem e desenvolvimento, nomeadamente LangChain. O LLM utilizado no projeto foi um LLM local para garantir a conformidade com o RGPD. Para isso foi utilizada a ferramenta Ollama, que permite a utilização de LLM localmente através de uma API na qual a aplicação pode enviar e receber mensagens processadas pelo LLM. Para ter acesso a conteúdos específicos, como a organização da disciplina e a literatura usada, a técnica RAG é utilizada através de uma base de dados vetorial, que permite a busca em linguagem natural do conteúdo necessário para aumentar o conhecimento do LLM. Esse recurso preenche algumas lacunas dos LLM, pois os treinados com dados globalmente disponíveis e do passado não têm conhecimentos específicos de informações novas, de livros e nem de conteúdos organizacionais da disciplina, como por exemplo datas de avaliações.

Além desses dados estáticos, o agente também tem acesso a dados dinâmicos, ou seja, dados do Moodle LMS (o ambiente virtual de aprendizagem utilizado). O acesso a esses dados é feito por via da API dos Web Services do Moodle.

Para que também haja interação recorrente do agente, foi criado um sistema chamado de ProactiveEngine, no qual o sistema possui alguns gatilhos configurados para que possa buscar novas informações e ficar atualizado com o que está a ocorrer na disciplina. Esse ProactiveEngine irá fazer uma chamada para buscar os dados do Moodle via API. Dessa forma o sistema está sempre atualizado com novas informações do que está ocorrendo na disciplina. Essa integração gera um desafio: a manipulação correta dos dados. Caso não haja a devida filtragem de dados, os pedidos na API se tornam exagerados (mais dados do que os necessários) e o sistema se torna uma réplica do Moodle LMS. Para evitar esses problemas, os dados que o Moodle pode passar são escolhidos de maneira a evitar o consumo de dados em excesso e é utilizado um sistema de resumo. O LLM usado no projeto assume também a função de resumir os dados do Moodle e qualificá-los, como por exemplo, identificar a intenção de um *post* num fórum. Isso faz com que os dados guardados na base de dados sejam resumidos (não sendo uma réplica do LMS) e também facilite ao LLM a tarefa de replicar a ação de um professor.

Para lidar com o RGPD de maneira adequada, o sistema corre em um ambiente local, assim sendo, a transmissão de dados ocorre entre o computador que executa o agente (que pode tanto ser um computador pessoal de um docente ou um computador da universidade) e o Moodle LMS diretamente via API, portanto, preservando a confidencialidade dos dados e sua transmissão apenas à máquinas da universidade ou dos seus funcionários, não a sistemas terceiros.

Para uma replicação fácil em múltiplos computadores e permitir com que o sistema seja utilizado pelos professores para terem seus “colegas codocentes”, o sistema foi pensado para se executar com auxílio do Docker e do Ollama (para se utilizar os LLM disponíveis publicamente).

4. Arquitetura e desenvolvimento

4.1. Arquitetura do sistema

O sistema é resumido pelo diagrama da Figura 1.

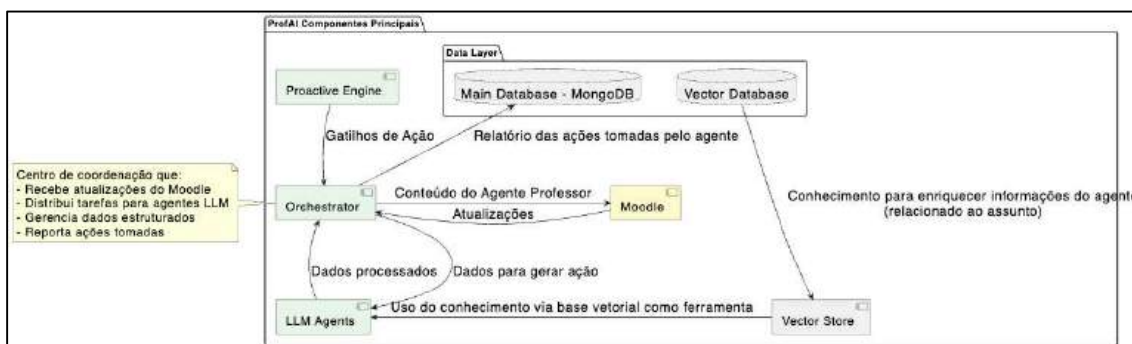


Figura 1. Diagrama de componentes

O Orchestrator é a peça central do sistema, o componente que intermedeia tudo o que ocorre. Os “LLM Agents” são os componentes que providenciam a forma “pensante” do sistema e geram respostas para o Orchestrator. Os agentes têm várias funções: resumir conteúdo, inferir intenções a partir do conteúdo do Moodle (por exemplo: um *post* no fórum do Moodle pode ser uma pergunta sobre o calendário da disciplina, sendo necessário inferir que a intenção é obter esse calendário) e gerar uma resposta destinada a um elemento do Moodle (por exemplo: uma resposta à pergunta do exemplo anterior pode ter de ser dada como publicação no mesmo fórum, no mesmo *thread*, como resposta à publicação anterior, sendo o conteúdo a explicação de qual o calendário ou onde pode ser consultado).

A camada de dados contém duas bases de dados, uma vetorial que armazena o conteúdo da disciplina e a organização da mesma; e uma base de dados semi-estruturada que armazena o histórico das ações que foram tomadas pelo sistema e os dados que foram obtidos do Moodle. A escolha da base de dados vetorial deve-se à necessidade de responder a consultas feitas por RAG com linguagem natural, permitindo que um

documento PDF seja pré-processado anteriormente e os seus dados armazenados de forma acessível (linguagem natural). Isso permite com que um agente LLM possa consultá-lo em linguagem natural para dele obter o conhecimento necessário para gerar uma ação de resposta.

Este fluxo pode ser visto da seguinte forma: um aluno cria uma pergunta via *post* no fórum do Moodle, essa informação é passada pelo Orchestrator para a base de dados estruturada. O componente LLM Agent é chamado pelo Orchestrator para gerar uma ação (caso necessário) para aquele novo conteúdo obtido do Moodle. O LLM Agent busca informações relevantes nas bases de dados a que tem acesso. Essa busca de informações relevantes pode ser por palavras-chave numa consulta à base de dados vetorial, algo como “data prova outubro”, o que possivelmente irá resultar em conteúdo da base de dados vetorial. Esse resultado da busca na base de dados vetorial irá ser processado com o resto dos elementos presentes no LLM, que irá gerar uma resposta tanto com o conhecimento novo vindo da base de dados vetorial como o conhecimento já adquirido anteriormente via treinamento, e também o conteúdo do Moodle recebido. Com os dados processados, o LLM Agent pode gerar uma nova ação (ou não), a sua resposta irá definir se uma ação deve ser tomada ou não, o tipo de ação e também o conteúdo (caso a ação deva ser tomada). Caso uma ação deva ser tomada, o Orchestrator irá tomar essa ação e publicar o conteúdo no Moodle. Ele também armazena na base de dados principal a ação tomada. Os detalhes podem ser vistos no diagrama de sequência (Figura 2).

Há ainda um pré-processamento do conteúdo (feito pelo componente VectorStore) da disciplina via os PDF para serem armazenados na base de dados vetorial (Figura 1).

4.2. Tecnologias usadas

As tecnologias usadas para fazer com que a RAG funcionasse se basearam principalmente na framework LangChain e na ferramenta Ollama.

A framework LangChain permite a fácil integração de ferramentas. As ferramentas são funções extra a que a LLM pode ter acesso. Uma das ferramentas integradas no agente via LangChain é a busca de dados na base de dados vetorial. Isso ocorre de maneira que o LLM processa o *input* inicial, verifica as ferramentas a que tem acesso e se há alguma ferramenta que possa ser utilizada para aprimorar a sua resposta, essa ferramenta é chamada, o que nesse caso se traduz numa busca. Com o resultado da busca, o *input* para o LLM é acrescido desses dados extra.

Para uso da LLM localmente, ou seja, *on-premise*, foi usado o Ollama, uma ferramenta que permite correr modelos de LLM em máquina local. O LangChain possui fácil integração com Ollama, o que permitiu uma rápida prototipação da solução integrando essas ferramentas.

A base de dados vetorial utilizada foi o Qdrant. Tem como função principal neste projeto armazenar o conteúdo estático da disciplina, tal como datas já estipuladas e manuais usados durante o ano curricular. A base de dados vetorial permite o armazenamento que facilita a busca via linguagem natural, que é o que o LLM juntamente com as suas

ferramentas produz. Essa busca via linguagem natural é feita utilizando um algoritmo de similaridade, que retorna uma lista de possíveis respostas de acordo com a busca.

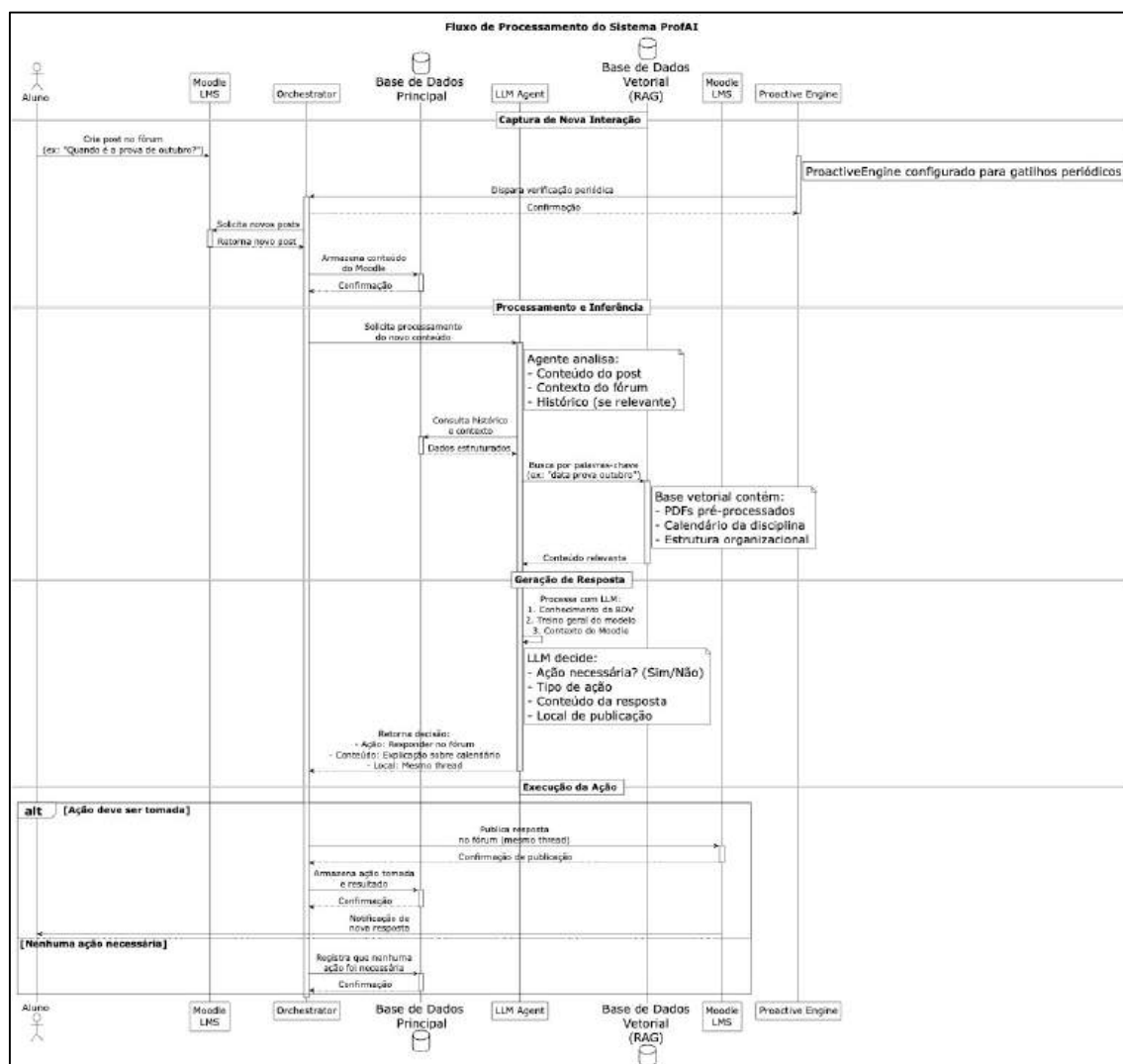


Figura 2. Diagrama de sequência - atualizações

Essas tecnologias foram integradas usando Docker, o que permitiu inicializar containers com as configurações necessárias para que a aplicação pudesse se comunicar com esses serviços como a LLM e as bases de dados vetorial e semi-estruturada (MongoDB). A linguagem de programação para que todo o projeto ganhasse vida foi Typescript com Node.js, viabilizando a comunicação entre todos os componentes do projeto.

4.3. Implementação prática

Para permitir com que o projeto funcionasse localmente e fosse possível testar a integração com o Moodle, foi necessário ter uma instância do Moodle local. Para isso também foi utilizado o Docker para criar um container e correr o Moodle localmente de uma maneira simples. Com o container a correr uma instância do Moodle foi possível configurar uma disciplina com dados fictícios e também configurar o Moodle Web

Services para expor alguns *endpoints*. Esses *endpoints* foram utilizados para comunicação com a aplicação via REST. Com a disciplina e o Moodle Web Services configurados foi possível interagir com o sistema via interface e também via pedidos HTTP (REST).

4.4. Segurança e privacidade

A segurança e a privacidade dos dados são pontos de extrema importância para o projeto, pensados desde a fase inicial. Para garantir que os dados são tratados de maneira correta sem a transmissão para terceiros, eles são processados em uma LLM local (LLM Agents – Figura 1) e armazenados localmente também em bases de dados locais no mesmo servidor (Data Layer –Figura 1).

Para garantir que o agente toma as ações adequadas e de que o processamento dos dados está sendo feito corretamente, relatórios de ações do agente são gerados. Esses relatórios contêm as informações das ações tomadas com seus conteúdos, a confiança daquela ação e metadados do Moodle, como por exemplo o *post* do fórum que foi respondido.

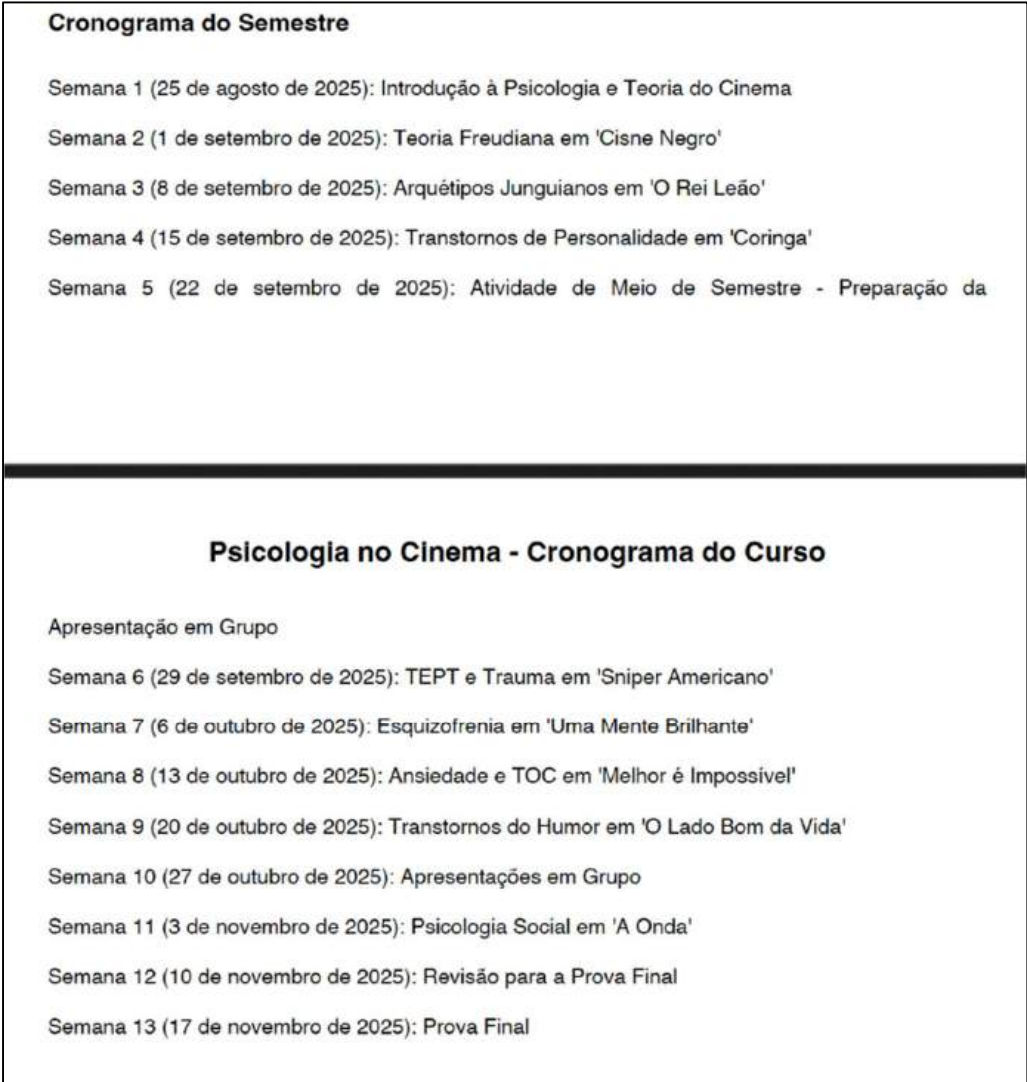


Figura 3. Cronograma da disciplina fictícia

5. Resultados

O projeto ProfAI teve como objetivo ser uma prova de conceito para ser replicada em ambiente de múltiplos codocentes na mesma instituição. A implementação da prova de conceito teve como foco atuar somente nos fóruns, sendo que a sua preparação consistiu em criar uma disciplina fictícia e utilizar um PDF com datas e informações fictícias da disciplina. Também foi usado um documento PDF como manual da disciplina, contendo conteúdo de um artigo relacionado a disciplina fictícia (“Psicologia no cinema”). A Figura 2 mostra o conteúdo do PDF representativo do cronograma da disciplina fictícia.

Sempre que identificava um novo *post* no fórum, o agente buscava informações em sua base de dados vetorial para conseguir gerar uma resposta. O processo, como explicado anteriormente, consistia em resumir e avaliar a intenção e logo após processar o que deveria ser feito.

Um exemplo de um *post* de um “aluno” (fictício) no fórum no qual pede informações da data de uma prova, o agente intitulado “PROF AI” responde após procurar dados na base de dados vetorial.



Figura 4. Exemplo pergunta-resposta ProfAI

Na Figura 4 é possível identificar a pergunta e a resposta. Depois da pergunta ser identificada pelo agente, o resumo é feito, como detalhado pelo *screenshot* da Figura 5.

```
Cron job started: Get updates, classify posts and generate actions
Last classified intent date: 2025-06-22T19:16:14.945Z
Retrieving updates since: 2025-06-22T19:17:14.945Z
Fetching module details from endpoint: http://localhost:8080/webService/rest/server.php?wstoken=mod_forum_get_discussion_post&moodlewsrestformat=json&postid=315
New updates retrieved: 1
Classified post: {
  userId: '2',
  courseId: '3',
  summarizedInput: '0 aluno solicita informação sobre a data final do exame da disciplina.',
  forumId: '315',
  postId: '315',
  intent: 'assignment_deadline_query',
  source: 'forum_post',
  createdAt: '2025-06-22T19:19:01.736Z',
  updatedAt: null
}
```

Figura 5. Logs de processamento e resposta

Depois do processo de resumir e identificar a intenção, esses dados são passados para o LLM novamente para gerar uma ação de resposta. Para que essas etapas ocorram de maneira correta, cada *input* contém o seu *system prompt* específico. A busca na base de dados vetorial pode ser vista no *screenshot* da Figura 6.

```
Getting actions to be taken...
Calling the getSubjectMetadata tool.
Getting subject metadata: exame da disciplina
Retrieving relevant knowledge for query: exame da disciplina
Retrieved documents: [
  Document {
    pageContent: 'Semana 12 (10 de novembro de 2025): Revisão para a Prova Final\n' +
      'Semana 13 (17 de novembro de 2025): Prova Final\n' +
      'Provas e Atividades\n' +
      '- Atividade de Meio de Semestre: Apresentação em Grupo em 27 de outubro de 2025\n' +
      '- Prova Final: Prova escrita abrangente em 17 de novembro de 2025\n' +
      '- Postagens semanais de discussão (no site do curso)\n' +
      '- Participação nas exposições de filmes e discussões em sala',
    metadata: {
      source: '/Users/gabriel/codes/college/projeto/profAI/src/KnowledgeBase/subject_metadata.pdf',
      pdf: [Object],
      loc: [Object]
    },
    id: 'f6a42e70-e09f-4756-ae1e-1d6bf5d00d56'
  },
  Document {
    pageContent: 'Psicologia no Cinema - Cronograma do Curso\n' +
      'Visão Geral do Curso\n' +
      'Este curso explora a interseção entre psicologia e cinema, analisando como as teorias e\n' +
      'transtornos psicológicos são representados nos filmes. Por meio de discussões críticas e exposições\n' +
      'de filmes, os alunos aprofundarão sua compreensão do comportamento humano, saúde mental e\n' +
      'expressão narrativa nas mídias visuais.\n' +
      'Informações da Professora\n' +
      'Professora: Dra. Laura Mendes\n' +
      'Email: laura.mendes@university.edu',
    metadata: {
      source: '/Users/gabriel/codes/college/projeto/profAI/src/KnowledgeBase/subject_metadata.pdf',
      pdf: [Object],
      loc: [Object]
    },
    id: '85c76643-7246-497b-8b95-722903c0d385'
  },
  Document {
    pageContent: 'Email: laura.mendes@university.edu\n' +
      'Horário de Atendimento: Terças e Quintas, das 14:00 às 16:00, Sala 204, Departamento de\n' +
      'Psicologia\n' +
      'Materiais do Curso\n' +
      '- Livro Obrigatório: "Filmes e a Mente: Teorias da Psicologia no Cinema" de Alan Thomas\n' +
      '- Exposições semanais de filmes (títulos serão anunciados com uma semana de antecedência)\n' +
      '- Artigos acadêmicos (disponíveis no site do curso)\n' +
      'Cronograma do Semestre\n' +
      'Semana 1 (25 de agosto de 2025): Introdução à Psicologia e Teoria do Cinema',
    metadata: {
      source: '/Users/gabriel/codes/college/projeto/profAI/src/KnowledgeBase/subject_metadata.pdf',
      pdf: [Object],
      loc: [Object]
    },
    id: 'fc8e1a08-9a5d-4395-b4a9-0b87efd27195'
  },
  Document {
    pageContent: 'Semana 2 (1 de setembro de 2025): Teoria Freudiana em "Cisne Negro"\n' +
      'Semana 3 (8 de setembro de 2025): Arquétipos Junguianos em "O Rei Leão"\n' +
      'Semana 4 (15 de setembro de 2025): Transtornos de Personalidade em "Coringa"\n' +
      'Semana 5 (22 de setembro de 2025): Atividade de Meio de Semestre - Preparação da',
    metadata: {
      source: '/Users/gabriel/codes/college/projeto/profAI/src/KnowledgeBase/subject_metadata.pdf',
      pdf: [Object]
    }
  }
]
```

Figura 6. Logs da busca vetorial

Depois disso uma ação de resposta é gerada e a ação em si é tomada, criando uma chamada POST ao Moodle com o conteúdo gerado (Figura 7).



Figura 7. Resposta do agente

Os detalhes da ação tomado pelo agente são armazenados da forma indicada na Figura 8.

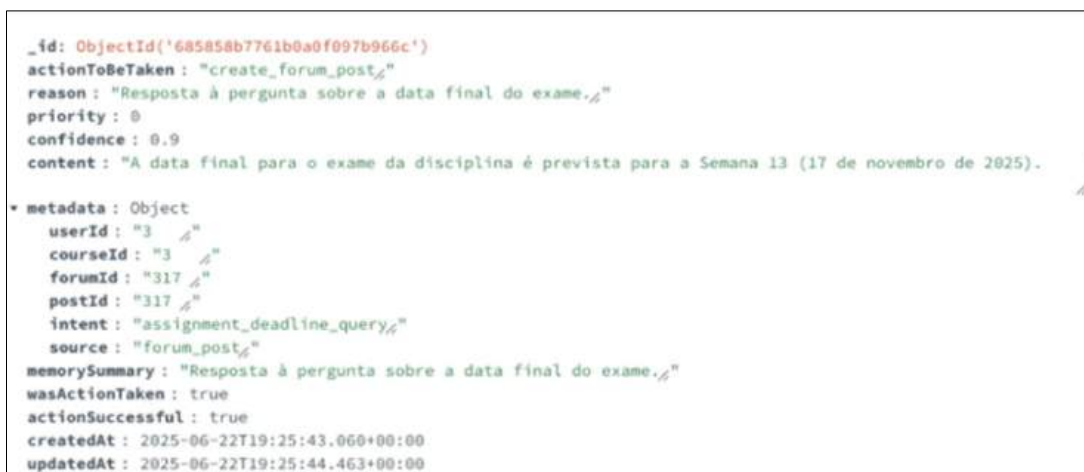


Figura 8. Report de ação do agente na base de dados

6. Discussão

Na secção 3 alguns problemas foram levantados para serem solucionados pelo projeto. Podemos identificar que alguns deles foram resolvidos em prova de conceito, como

redução do trabalho manual e da necessidade de orquestração docente. Esse problema foi resolvido através do resumo e classificação de intenção dos conteúdos, o que geraria um grande trabalho manual do docente para interpretar cada *post*. Além disso o docente iria ter de elaborar uma resposta correta e encontrar as informações necessárias para a resposta, o que o agente já consegue resolver.

Um outro problema citado foi a falta de personalização e contextualização de *feedback* feito por *chatbots* comuns. O agente ProfAI pode ser configurado para o conteúdo necessário, bastando apenas providenciar os dados para ele, inserindo o conteúdo em sua base de dados vetorial. Essa fonte de conhecimento permite que ele obtenha o conhecimento personalizado para cada disciplina, permitindo sua replicabilidade e configuração para disciplinas diferentes.

Alguns problemas, entretanto, não foram totalmente resolvidos. Em alguns casos o agente não conseguia responder corretamente as perguntas ou alucinava, gerando respostas aleatórias. É provável que esse tipo de problema possa ser minorado com modelos de LLM com mais parâmetros, visto que o modelo utilizado (llama3-groq-tool-use) foi um modelo bem inferior aos modelos com melhores resultados com apenas 8B de parâmetros (4.7Gb). Modelos de alto desempenho com resultados melhores em *benchmarks* têm em média 43B de parâmetros (43Gb). O teste com modelos melhores não foi possível devido a limitação computacional no qual o computador que estava a executar a aplicação possuía. Ainda assim, note-se que as divergências de contexto ou “alucinações” são um problema recorrente nos LLM, embora os sistemas online comerciais tenham vindo a demonstrar que pode ser consideravelmente limitado.

Outro problema que não foi resolvido devido a ser uma prova de conceito foi a generalização de casos, ou seja, apenas o fórum foi alvo de desenvolvimento para essa prova de conceito e não mensagens diretas com alunos, questionários, etc.

7. Conclusão e Trabalhos Futuros

Em termos gerais, o ProfAI conseguiu atingir os resultados esperados, que eram de demonstrar o conceito de que é possível proporcionar *feedback* individualizado em contextos de e-learning com grandes números de estudantes, através de uma arquitetura RAG com acessos a recursos relacionados a disciplina a ser lecionada.

A execução local, *on-premises*, também demonstrou ser possível fazê-lo em cumprimento do RGPD, permitindo com que os dados fossem tratados localmente pela universidade ou por computadores relacionados a docentes da universidade. Apesar desses resultados terem sido atingidos, o projeto ainda é uma prova conceitual, sendo relevante efetuar uma avaliação mais aprofundada e em ambiente real.

Como citado na secção 5, o modelo usado ainda gera respostas incorretas em alguns casos e permite com que qualquer *input* de informação seja tratado. Uma melhoria importante ao projeto seria criar filtros para impedir o processamento de *inputs* que não deveriam ser

tratados. Juntamente com essa melhoria, um *fine-tuning* dos modelos também pode ser feito para tentar otimizar as respostas de acordo com o ambiente em que está sendo utilizado (um ambiente virtual de aprendizagem). Mas não dispensa o registo de regras administrativas e processuais, regulamentos, que preconizam formas de atuação e resolução nestas situações.

Há ainda uma configuração manual necessária a ser feita pelo docente, diretamente via ficheiros de configuração. A implementação de uma interface para que o docente faça a configuração do projeto é também algo essencial.

A continuação do projeto também passa por analisar o quão efetivo o mesmo está sendo. Para isso é preciso criar métricas para analisar o tempo economizado do docente, a qualidade do conteúdo gerado e a satisfação dos alunos em relação ao ProfAI.

Um trabalho futuro de continuação deste projeto poderá ser anonimizar os dados dos docentes e alunos, permitindo o recurso a modelos mais poderosos, em instalações exteriores à universidade. O processo de entendimento por síntese e não dados em bruto já foi feito, esse processo pode ser usado de maneira similar para efetuar a anonimização dos dados.

Em síntese, o ProfAI representa um avanço significativo no caminho para agentes codocentes éticos e replicáveis. Resolveu de forma consistente os desafios da automatização e privacidade, mas requer validação empírica, aprimoramento pedagógico e autorregulação adaptativa para atingir maturidade plena. O trabalho futuro deve concentrar-se em testar, medir e refinar o impacto do ProfAI como mediador inteligente na aprendizagem digital.

REFERÊNCIAS

- Holstein K., McLaren B. M., and Aleven V. (2019). Co-Designing a Real-Time Classroom Orchestration Tool to Support Teacher-AI Complementarity, *Journal of Learning Analytics*, vol. 6, no. 2, pp. 27-52, <https://doi.org/10.18608/jla.2019.62.3>.
- Prieto, L. P., Rodríguez-Triana, M. J., Martínez-Maldonado, R., Dimitriadis, Y., & Gašević, D. (2019). Orchestrating learning analytics (OrLA): Supporting inter-stakeholder communication about adoption of learning analytics at the classroom level. *Australasian Journal of Educational Technology*, 35(4). <https://doi.org/10.14742/ajet.4314>
- Maier U. and Klotz C. (2022). Personalized Feedback in Digital Learning Environments: Classification Framework and Literature Review, *Computers and Education: Artificial Intelligence*, vol. 3, article 100080, <https://doi.org/10.1016/j.caeai.2022.100080>.
- Kuhail M.A., Alturki N., Alramlawi S., and Alhejori K. (2022). Interacting with educational chatbots: A systematic review, *Education and Information Technologies*, vol. 28, no. 1, pp. 973-1018, <https://doi.org/10.1007/s10639-022-11177-3>.

Labadze L., Grigolia M., and Machaidze L. (2023). Role of AI chatbots in education: systematic literature review, *International Journal of Educational Technology in Higher Education*, vol. 20, article 56, <https://doi.org/10.1186/s41239-023-00426-1>.

Adamopoulou E. and Moussiades L. (2020). Chatbots: History, technology, and applications, *Machine Learning with Applications*, vol. 2, article 100006, <https://doi.org/10.1016/j.mlwa.2020.100006>.

Winkler R. and Söllner M. (2018). Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis, in *Academy of Management Annual Meeting (AOM)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, <https://doi.org/10.48550/arXiv.1706.03762>.

Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson. ISBN: 978-1292401133.

Thorat S.A. and Jadhav V.D. (2020). A Review on Implementation Issues of Rule-based Chatbot Systems, in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, <http://dx.doi.org/10.2139/ssrn.3567047>.

Shulman L.S. (1986). Those Who Understand: Knowledge Growth in Teaching, *Educational Researcher*, vol. 15, no. 2, pp. 4-14, <https://doi.org/10.3102/0013189X015002004>.

OpenAI. (2024). Learning to reason with LLMs, <https://openai.com/index/learning-to-reason-with-llms>.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Thoeni, A., & Fryer, L. K. (2025). AI Tutors in Higher Education: Comparing Expectations to Evidence, https://doi.org/10.31219/osf.io/24tg7_v1.



Gabriel Aragão é Licenciado em Engenharia Informática (2025) pela Universidade Aberta (UAb). Atualmente é Software Developer, na empresa Volkswagen Group Digital Solutions [Portugal], onde desenvolve aplicações web. Tem como áreas de interesse Inteligência Artificial Generativa e desenvolvimento de aplicações escaláveis.



Leonel Morgado é Professor Catedrático da Universidade Aberta (UAb) e Investigador Coordenador no INESC TEC, cujo Núcleo coordena na UAb. Adido para Normas da Comissão Técnica para Ambientes de Aprendizagem Imersivos da IEEE Education Society, membro da Comissão Diretiva e do Círculo de Académicos da Immersive Learning Research Network. Foi diretor técnico-comercial de uma empresa de importação, distribuição e venda de hardware, terminologista da localização do Microsoft Office 97 e do Oracle InterOffice, consultor linguístico da IBM/Lotus, e coordenador de equipas de desenvolvimento Web, implantação de software e de combate à infoexclusão.