



## Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2025/2026  
2º Semestre

Enunciado do Trabalho Prático

### Utilização de LLM no Desenvolvimento de Software Aplicacional

2026, fevereiro

#### 1 Enquadramento

Durante os últimos anos, o domínio da Engenharia de Software tem enfrentado uma revolução séria na forma como deve ser aplicada e desenvolvida. O aumento da complexidade dos sistemas de software, a consolidação de arquiteturas distribuídas, a adoção generalizada de práticas de desenvolvimento e a crescente integração de técnicas de Inteligência Artificial no processo de software têm transformado profundamente a forma como as práticas de análise, a especificação e o desenvolvimento de software são aplicadas na resolução de problemas do mundo real contemporâneo. O software deixou de ser apenas um artefacto estático, passando a assumir um papel central em sistemas dinâmicos, em evolução permanente, e fortemente orientados por dados. Hoje o software tem de ter a capacidade de se adaptar de forma contínua, ao longo do tempo, conforme os processos vão evoluindo e as necessidades dos seus utilizadores vão aumentando. Quase ao mesmo tempo, a emergência e o desenvolvimento dos *Large Language Models* (LLM) mudaram a forma, os métodos e as técnicas, como o software é idealizado, especificado, desenvolvido, validado e mantido. Os LLM são treinados com grandes volumes de conhecimento, que lhes permitem apresentar capacidades bastante evoluídas em muitas atividades do mundo real. Também, no domínio da Engenharia de Software, a sua atividade (e impacto) começa a ser notada, sendo referenciados frequentemente como agentes artificiais com capacidades para colaborar de forma ativa nas várias tarefas que usualmente se desenvolvem ao longo do ciclo de vida do desenvolvimento de software. Hoje, a sua aplicação vai já para além da simples geração de código, abrangendo também muitas das tarefas da Engenharia de Requisitos, da definição e caracterização arquitetural, da realização de processos de validação e de teste, e no suporte à tomada de decisão numa grande diversidade de aspetos técnicos. A incorporação de LLM no ciclo de vida do desenvolvimento de software antecipa um futuro no qual os engenheiros de software passam a desempenhar um papel cada vez mais relevante, estratégico, focado na definição de problemas, na avaliação crítica de soluções e na governação de sistemas inteligentes. As tarefas repetitivas ou cognitivamente intensivas estão a ser progressivamente delegadas para os agentes artificiais, que

apliquem, direta ou indiretamente, técnicas e modelos de *Inteligência Artificial Generativa* (IAG). Este cenário coloca novos desafios técnicos, metodológicos, sociais e éticos, mas abre também grandes oportunidades para as empresas e os seus profissionais obterem ganhos de produtividade, na qualidade e na inovação e, obviamente, reduzirem significativamente o seu “time-to-market”.

## 2 O Trabalho Prático

O trabalho prático da unidade curricular de Laboratórios de Informática IV (LI4), para o presente ano letivo, envolve a adoção de uma nova abordagem no processo tradicional do desenvolvimento de software. De forma a acompanhar as novas tendências de atuação dos profissionais e garantir um acompanhamento sustentado da evolução e aplicação dos LLM no domínio da Engenharia de Software, o trabalho de LI4 será desenvolvido de forma assistida por agentes artificiais, explorando de forma sistemática o uso de LLM ao longo de todas as fases do ciclo de vida do desenvolvimento de software. O trabalho foi idealizado especificamente para que os alunos da licenciatura em Engenharia Informática possam conceber, especificar, desenvolver, validar e evoluir um sistema de software utilizando LLM, como agentes artificiais, de forma sistemática, em todas as fases do ciclo de vida do desenvolvimento de software. O desenvolvimento deste trabalho seguirá uma metodologia sequencial e incremental, inspirada em modelos clássicos como Waterfall (Royce, 1970) ou V-Model (Forsberg et al., 2005), que será enriquecida com práticas modernas de desenvolvimento de software – e.g. DevOps (Leite et al., 2019), MLOps (Sculley et al., 2015) e AI-assisted SE (Chen et al., 2021).

O trabalho prático será desenvolvido com base num conjunto de temas específicos, envolvendo aplicações de processamento de dados, consideradas bastante tradicionais. Os temas que foram escolhidos para a presente edição da unidade curricular LI4 foram os seguintes:

- **Tema 1** – *Sistema de gestão integrada para uma cadeia de pequenas lojas de conveniência*. O sistema de software a desenvolver neste tema terá como objetivo apoiar a gestão de uma cadeia de pequenas lojas de conveniência, através da implementação de meios e estruturas adequadas para o registo e processamento centralizado dos dados das vendas, produtos, stocks e fornecedores, de cada uma das lojas. Deve-se ter em conta que, cada uma das lojas da cadeia atua de forma autónoma, mas saber também que a sua informação deverá ser, todos os dias, ao fim do dia, consolidada num sistema central de dados para suporte a processos de análise de dados. O sistema a implementar deverá disponibilizar serviços para a gestão e controlo de inventário, reposição de produtos nas lojas, faturação e geração de relatórios de gestão, por loja, por período e por categoria de produto, de forma que seja possível apoiar adequadamente os processos de tomada de decisão e operacionais da cadeia de lojas.
- **Tema 2** – *Sistema de gestão para um clube desportivo*. Um sistema destinado à gestão de clubes desportivos, academias ou associações, que permita fazer o registo de atletas, equipas, treinadores e

modalidades. O sistema deverá permitir controlar as inscrições, as presenças em treinos, as participações em competições, bem como os seus resultados, disponibilizar estatísticas de desempenho e relatórios financeiros relativos às quotas ou pagamentos realizados. Além disso, deve ser também capaz de produzir relatórios desportivos e administrativos, que possuam informação pertinente para apoiar os processos de tomada de decisão desenvolvidos pela direção técnica do clube.

- **Tema 3 – *Sistema de gestão para uma oficina de reparação de trotinetes*.** O sistema deverá ser capaz de apoiar os processos de gestão de uma oficina de reparação de trotinetes, através da disponibilização de serviços para fazer o registo de clientes, equipamentos, ordens de serviço e intervenções realizadas nas trotinetes. Além disso, o sistema deverá controlar os processos de diagnóstico realizados sobre as trotinetes, bem como tratar de todo o processo de reparação de uma trotinete, permitindo o registo das peças utilizadas, os tempos de reparação, os estados do serviço e, obviamente, da relação dos custos associados com a reparação. Por fim, o sistema deve ser também capaz de fazer a gestão do stock de peças, a faturação dos serviços prestados e peças comercializadas e a geração de relatórios técnicos e financeiros, para facilitar o acompanhamento das operações e os processos de tomada de decisão realizados pela gerência da oficina.
- **Tema 4 – *Sistema de gestão para um hotel de animais*.** O sistema tem como objetivo apoiar a gestão de um hotel de animais, disponibilizando serviços específicos para fazer o registo dos animais, reservas e períodos de estadia, bem como acolher os dados dos seus proprietários. O sistema deverá também ser capaz de controlar os serviços associados com as estadias dos animais, como seja a alimentação, os cuidados veterinários, os banhos ou os passeios, e a disponibilidade dos espaços de acolhimento que o hotel disponibiliza. Além disso, deverá também permitir fazer a gestão de pagamentos, registar o histórico das estadias e fazer a geração de relatórios operacionais, de forma que seja possível fazer uma organização eficiente e o acompanhamento adequado de cada animal que fique no hotel.

### 3 As Etapas do Trabalho

O trabalho prático foi estruturado com base num conjunto de etapas sequenciais, interligadas, que deverão ser realizadas durante um período de cerca de cinco meses, definindo a forma como o processo de desenvolvimento do software será realizado. Genericamente, as etapas definidas abrangem todo o ciclo de desenvolvimento de software, desde o levantamento e análise de requisitos até à implementação do sistema definido, bem como a definição e aplicação de testes e preparação do software para entrega. Para cada uma das etapas foram definidos um conjunto de objetivos bem definidos e uma relação dos resultados esperados, que satisfaçam os requisitos operacionais da etapa seguinte. Este tipo de abordagem permite ter um maior controle sobre o processo de desenvolvimento, como também permite garantir um nível de qualidade de trabalho elevado e alinhar o desenvolvimento do sistema com as necessidades definidas, no início e ao longo da realização de cada uma das etapas. As etapas que foram definidas são as seguintes:

- **Etapa 1 — *Conceção e Engenharia de Requisitos Assistida por LLM***. Nesta etapa pretende-se definir claramente o problema e o domínio de aplicação do sistema, introduzir os LLM como agentes de apoio cognitivo ao processo de Engenharia de Requisitos e produzir a documentação formal de requisitos necessária, de forma estruturada e alinhada com as normas reconhecidas, para servir como base as etapas seguintes do projeto.
  - Tarefas: Definição do domínio do sistema, identificação de *stakeholders*, análise de contexto e restrições, eliciação de requisitos, geração de *user stories*, refinamento de requisitos ambíguos, simulação de entrevistas com *stakeholders*, requisitos funcionais e não funcionais, especificação do software (SRS – IEEE 830/29148), casos de uso e diagramas UML.
  - Período de Realização: 09FEV2026-02MAR2026 (3 semanas)
  
- **Etapa 2 — *Arquitetura e Design do Software utilizando LLM***. Nesta parte do projeto pretende-se definir a arquitetura global do sistema e os seus principais componentes, explorar o uso de LLM como agentes de apoio à tomada de decisões arquiteturais e ao projeto (*design*) de software, especificar o software a desenvolver e produzir a documentação técnica necessária, para descrever a estrutura, o comportamento e as interfaces do sistema, com vista à sua posterior implementação.
  - Tarefas: Definição da arquitetura global do sistema, explorar LLM como assistentes de projeto e de revisão arquitetural, desenvolver os diagramas de classes, sequência e componentes necessários, fazer o *design* de interfaces, e documentar a arquitetura definida para o software, incluindo o diagramas UML produzidos, a especificação de API e as decisões arquiteturais tomadas.
  - Período de Realização: 02MAR2026-23MAR2026 (3 semanas)
  
- **Etapa 3 — *Implementação e Desenvolvimento Assistido por LLM***. Na terceira etapa do projeto pretende-se fazer a implementação do sistema através da codificação dos seus diversos componentes, utilizar LLM como agentes de apoio na geração, fazer a revisão e reestruturação (se necessário) do código produzido, assegurar que o software desenvolvido cumpre os requisitos e padrões de qualidade previamente definidos, e produzir a documentação necessária para as etapas seguintes de verificação e validação do trabalho desenvolvido.
  - Tarefas: Configuração do ambiente de trabalho, implementação incremental do software pretendido, desenvolvimento por módulos, geração de código assistida por LLM, revisão

automática e humana, revisão do código gerado (*code reviewing*), deteção de *smells* e sugestão de modificações (*refactoring*).

- Período de Realização: 23MAR2026-13ABR2026 (3 semanas)
- **Etapa 4 — *Verificação, Validação e Avaliação da Qualidade do Software Produzido***. Por fim, nesta etapa final, pretende-se desenvolver trabalho específico para garantir que o sistema implementado cumpre os requisitos especificados e atende aos padrões de qualidade esperados, utilizar LLM como agentes de apoio à geração de casos de teste, na análise de cobertura e na deteção de inconsistências, e produzir relatórios de teste e métricas de qualidade, que permitam avaliar a confiança, robustez e adequação do software antes da sua instalação e entrada em produção (*deployment*).
  - Tarefas: Realização de testes automatizados, unitários, de integração e de sistema, geração de testes com LLM (e.g. *test cases*, *edge cases* e testes baseados em requisitos), verificação da satisfação dos SRS (*Software Requirements Specification*), e realização de testes de aceitação, com análise de qualidade, cobertura de código e métricas de qualidade (ISO/IEC 25010).
  - Período de Realização: 13ABR2026-11MAI2026 (4 semanas)

Após a etapa 4, no final do processo de desenvolvimento, os alunos deverão preparar o sistema para instalação operacional, desenvolver um guia para a sua operação e manutenção, elaborar o relatório final do projeto e preparar uma apresentação técnica aos docentes da unidade curricular sobre o trabalho realizado e o software produzido.

Período de Realização: 26MAI2026-27MAI2026 (3 dias).

#### 4 Resultados Esperados

O presente trabalho tem como objetivo proporcionar aos alunos de Engenharia Informática uma experiência abrangente e prática no domínio da Engenharia de Software, alinhada com os padrões de conceção e desenvolvimento de projetos de software por profissionais, que lhes permita desenvolver as suas competências técnicas e sustentar e desenvolver críticas de trabalho (operacionais e analíticas). No final do projeto, espera-se que os alunos sejam capazes de:

- Realizar e gerir todas as fases do ciclo de vida de desenvolvimento de software, desde a sua conceção e análise de requisitos, até à implementação, testes, integração e manutenção de software.
- Compreender o papel dos LLM quando integrados como agentes inteligentes dentro do processo de desenvolvimento de software, automatizando tarefas de desenvolvimento rotineiras e padronizadas e fornecendo suporte à tomada de decisão ao longo das diversas etapas do ciclo de vida do

desenvolvimento de software.

- Identificar cenários de aplicação apropriados e projetar soluções que aproveitem técnicas e modelos de IAG de forma prática, produtiva, eficaz e, obviamente, ética.
- Avaliar os benefícios e limitações da Inteligência Artificial na Engenharia de Software, desenvolvendo uma visão crítica sobre o uso da IAG, tendo em conta os seus benefícios e limitações.
- Saber produzir peças de software comparáveis a projetos de nível profissional, apresentando código-fonte estruturado, documentação completa, testes automatizados e protótipos com funcionalidades avançadas.

Em suma, neste trabalho procuramos criar e desenvolver um meio de aprendizagem avançado para os alunos no domínio da Engenharia de Software, que combina a aplicação de teoria, prática e reflexão crítica, visando a sua preparação para atuar de forma competente e ética no desenvolvimento de soluções de software que envolvam IAG, consolidando habilidades técnicas e analíticas comparáveis às exigidas no mercado profissional.

## 5 Comentário Final

A integração de LLM no domínio da Engenharia de Software tem contribuído com inúmeros aspetos inovadores, que permitem aumentar a produtividade das equipas de desenvolvimento, melhorar a qualidade do código produzido e automatizar tarefas repetitivas e complexas. Os LLM podem desenvolver trabalho em várias vertentes do processo de desenvolvimento de software. De referir: a geração e revisão de código-fonte, documentação automática, suporte à análise de requisitos e projeto dos sistemas de software, e testes e validação do software produzido. Porém, devemos ter sempre presente e ter consciência de que a integração eficaz de LLM requer conhecimento sólido dos fundamentos, técnicas, modelos e práticas da Engenharia de Software, bem como possuir uma experiência robusta na conceção, modelação, definição de arquiteturas, integração de componentes e boas práticas de codificação, entre outras coisas mais. Sem que isso se verifique, a utilização de LLM pode produzir soluções inadequadas, mal construídas e fundamentadas, e difíceis de manter. Além disso, devemos ter em conta que também é fundamental implementar mecanismos de governação, validação e responsabilidade técnica ao longo do processo de desenvolvimento de software assistido por LLM. Em suma, num processo de desenvolvimento de software, os LLM deverão funcionar como agentes de suporte inteligente, cuja utilização adequada e com sucesso dependerá de engenheiros com conhecimento e perícia adquirida nesse tipo de processos, que sejam capazes de aplicar práticas de governação e validação rigorosas. Essa integração equilibrada permitirá aumentar a produtividade dos profissionais, melhorar a qualidade do software desenvolvido e mitigar eventuais riscos associados com a automação avançada da produção de software.

## 5 Bibliografia

Como suporte à realização deste trabalho os grupos deverão apoiar-se nas referências bibliográficas recomendadas pelo docente responsável pela unidade curricular ao longo da sua leção. De forma complementar recomenda-se, também, a consulta ou leitura das seguintes publicações:

- ISO/IEC/IEEE 12207 : 2017(E) First Edition 2017-11 - Redline: ISO/IEC/IEEE International Standard - Systems and Software Engineering -- Software Life Cycle Processes - Redline / Institute of Electrical and Electronics Engineers. IEEE; 2017. <https://www.iso.org/obp/ui/en/#iso:std:iso-iec-ieee:12207:ed-1:v1:en>
- Royce, W. W. (1970). Managing the development of large software systems. Proceedings of IEEE WESCON, 26(8), 1–9. <https://doi.org/10.1109/WESCON.1970.4954670>
- Forsberg, K., Mooz, H., & Cotterman, H. (2005). Visualizing project management (3rd ed.). John Wiley & Sons.
- Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.
- Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.
- Leite, L., Rocha, C., Kon, F., Milojevic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. ACM Computing Surveys, 52(6), Article 127. <https://doi.org/10.1145/3359981>
- Kim, G., Debois, P., Willis, J., & Humble, J. (2016). The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations. IT Revolution Press.
- Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley.
- Sculley, D., Holt, G., Golovin, D., et al. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems (NeurIPS), 28, 2503–2511.
- Feldt, R., Torkar, R., Afzal, W., et al. (2018). Artificial intelligence in software engineering: A systematic mapping study. IEEE Transactions on Software Engineering, 44(6), 1–31. <https://doi.org/10.1109/TSE.2018.2793609>
- Chen, J., Wang, S., Chen, X., et al. (2021). Software engineering for AI-based systems: A survey. ACM Transactions on Software Engineering and Methodology, 31(2), Article 26. <https://doi.org/10.1145/3472810>

\* \* \* \* \*