Plano da Unidade Curricular

Sítio:	PlataformAbERTA	Impresso por:	Carlos Teixeira
Unidade curricular:	Laboratório de Desenvolvimento de Software 2023 (Espaço central)	Data:	Quinta, 7 Março 2024, 22:24
Livro:	Plano da Unidade Curricular		

Descrição

Documento com o PUC desta unidade curricular.

Índice

PUC- Plano da Unidade Curricular

PUC

1. A Unidade Curricular

2. Competências

3. Roteiro

4. Metodologia
4.1. Estrutura das equipas da SimProgramming

5. Recursos

6. Utilização de ferramentas de inteligência artificial

7. Avaliação
7.1. Cartão de Aprendizagem
7.2. Calendário de avaliação contínua
7.3. Exame

8. Plano de Trabalho

PUC- Plano da Unidade Curricular

PUC - PLANO DE UNIDADE CURRICULAR

Laboratório de Desenvolvimento de Software - 21179
Docente: Leonel Morgado Tutor: Tiago Romão
Ano Letivo: 2023/2024

PUC

O que é o PUC?

O Plano da Unidade Curricular (PUC) constitui um documento que visa orientar o processo de aprendizagem do estudante ao longo da Unidade Curricular a que se refere. É por isso um guião que requer uma leitura atenta, que lhe será útil ao longo de todo o percurso de aprendizagem. Aqui encontrará informação sobre as temáticas a estudar, as competências a desenvolver, informações sobre como se organiza o processo de aprendizagem, como utilizar e tirar partido do espaço virtual relativo a esta Unidade Curricular, o que se espera de si, como é avaliado, entre outros aspetos fundamentais para realizar da melhor forma este percurso.

1. A Unidade Curricular

Apresentação da Unidade Curricular

Esta unidade curricular fornece aos estudantes um contacto próximo com ambientes, ferramentas e métodos de desenvolvimento de software, apoiando a transição de projetos individuais para projetos em equipa, em maior escala.

O nosso objetivo não é conseguir desenvolver software, é encontrar a melhor maneira de o fazer, de forma a minimizar os custos e maximizar a qualidade do software, e - porque não - maximizar o prazer de o desenvolver.

Adota-se uma abordagem didática diferente do que é habitual: ***e-SimProgramming***.
(Quem tiver curiosidade pode ler o artigo-resumo: a investigação premiada sobre didática de engenharia de software que lhe deu origem.)

Nesta abordagem, os alunos de engenharia de software são imersos numa narrativa interativa, onde assumem o papel de estagiários recém-chegados à empresa "SimProgramming". Dentro dessa narrativa, irão elaborar as várias atividades formativas e de avaliação (e-fólios).

2. Competências

Competências a desenvolver

Os alunos deverão adquirir o domínio de conceitos de engenharia de software que suportam os objetivos da unidade curricular:

- Capacidade para analisar um problema de desenvolvimento e estruturá-lo numa arquitetura modular;

- Capacidade para definir e implementar a arquitetura modular com abordagens de código que revelem as qualidades desejadas para um software que seja uma peça de engenharia: simplicidade, independência para fomentar a reutilização, acoplamento fraco com coesão forte, entre outras;

- Capacidade para tratar erros de forma independente do componente onde ocorrem e respeitando a arquitetura modular.

3. Roteiro

Roteiro do conteúdo a trabalhar

Nesta Unidade Curricular serão trabalhadas as seguintes temáticas :

1. Conceitos introdutórios:
    - Software artesanal vs. Engenharia de software
    - Diferenças entre C++ e C# necessárias à boa interpretação dos exemplos
    - Utilização do ambiente integrado de desenvolvimento Visual Studio

2. Princípio do design de software:
    - Os estilos arquitetónicos do software
    - O estilo arquitetónico Model-View-Controller (MVC)
        - Variante com input no Controller
        - Variante com input na View
        - Variante mista

3. Separação de interesses
    - Conceito
    - Atributos de qualidade do software
    - Abordagens de implementação com código em MVC

4. Testes de software e tratamento de erros
    - Tipos de testes de software
    - Tratamento de erros com exceções no contexto de MVC

5. Separação entre interface (API e visual) e implementação
    - Conceito
    - Abordagens de implementação com código em MVC

6. Melhoria final dos projetos

4. Metodologia

Como vamos trabalhar?

A unidade curricular divide-se em seis módulos. Em cada módulo o estudante tem ao seu dispor um conjunto de materiais online: capítulos dos livros recomendados e outras fontes de leitura, bem como vídeos e/ou diaporamas. Há livros opcionais que podem ser lidos ou consultados para complementar os materiais disponibilizados online.

e-SimProgramming: uma abordagem pedagógica

A unidade curricular segue a abordagem pedagógica e-SimProgramming (quem tiver curiosidade pode ler o resumo da investigação didática premiada que a originou).

Nesta abordagem pedagógica, tudo decorre no âmbito de uma narrativa interativa, onde os alunos são estagiários recém-chegados à empresa SimProgramming, na companhia de personagens virtuais que são colegas da empresa. O professor e eventuais tutores assumem papéis de diretor-executivo ou gestores de projetos, podendo ainda palestrantes convidados surgir com outros papéis nesta narrativa.

Atividades

As atividades têm como objetivo a concretização dos conceitos teóricos e das suas consequências para a prática de desenvolvimento de software, permitindo ao aluno constatar a existência de dúvidas ou diferenças de interpretação da matéria. São um recurso essencial à aquisição do conhecimento, e devem ser realizadas em combinação com a leitura dos materiais respetivos.

Cada módulo tem atividades de avaliação contínua, que visam proporcionar compreensão dos conceitos teóricos numa perspetiva de aplicação prática.

Exceto no primeiro módulo, caso o estudante tenha optado pela avaliação contínua, tem ao longo do semestre de ir preservando essas atividades enquanto portfólio digital, para apresentação em três pontos de avaliação quantitativa (e-fólios), onde se proporciona feedback quantitativo sobre as atividades realizadas até então. Ou seja, **não há um enunciado adicional de e-fólio**.

Repita-se: **não há um enunciado adicional de e-fólio** além das atividades formativas. (Então o que são os "e-fólios"? Portfólios. Reler o parágrafo anterior e o seguinte.)

As semanas de "e-fólio" são apenas semanas em que o docente corrige, para atribuição de nota, as atividades desenvolvidas até então. Até ao início das semanas de avaliação, os alunos podem melhorar ou corrigir os trabalhos em curso, com base no *feedback* do docente e dos colegas. Os resultados dos trabalhos em curso ficam disponíveis ao docente como portfólio, ou seja: repositório de código do projeto e conjunto de documentos produzidos e entregues nessas semanas.

Durante a realização de cada módulo há um fórum geral aberto, no qual os estudantes devem discutir a matéria e a abordagem à resolução das atividades, e o professor intervém assim que considere necessário. Um fórum de notícias e ajuda está aberto ao longo de todo o curso, de forma a ser utilizado para a publicação de notícias e dar apoio aos estudantes em questões que não sejam da matéria.

Flexibilidade Temporal

Ao longo de todo o curso são indicadas diversas datas. Estas são indicativas, podendo os alunos efetuar as atividades aquando da sua conveniência. **A única exceção são as datas de avaliação quantitativa (e-fólios)**. Nessa ocasião os portfólios serão avaliados pelo professor para conferir uma nota e as atividades precedentes terão de estar concluídas. Repita-se o que é dito acima: **não há enunciados adicionais de e-fólio além das atividades formativas**: as semanas de "e-fólio" são apenas as semanas em que o docente corrige, para atribuição de nota, as atividades desenvolvidas até então. Até ao início das semanas de avaliação, os alunos podem melhorar ou corrigir os trabalhos em curso, com base no *feedback* do docente e dos colegas.

Caso algum estudante não possa cumprir as datas de avaliação quantitativa indicadas, sugere-se a opção pela realização de exame, em vez da avaliação contínua.

Trabalho em equipa

Os alunos que optarem pela avaliação contínua deverão desenvolvê-la em grupo (equipa). As atividades deverão explicitar o contributo de cada membro para o resultado final, ou seja, a avaliação requer que seja explícita nos relatórios a coordenação interna da equipa e as responsabilidades individuais.

Este trabalho em equipa nasce do contexto da maioria dos projetos modernos de software, desenvolvidos em contextos de equipa. Desta forma vivenciarão melhor os contextos em que as técnicas estudadas se revelam úteis. Para tal, devem desde logo interagir com os colegas para identificar quem tem planos e abordagem de trabalho compatíveis. Para mais detalhes vejam os pormenores sobre esta forma de trabalho, em: "Estrutura das equipas da SimProgramming". O desenvolvimento individual será sempre excecional, após apreciação da fundamentação apresentada aos docentes.

4.1. Estrutura das equipas da SimProgramming

Caros estagiários da SimProgramming,

Cada equipa de estagiários deve distribuir responsabilidades por três papéis: **líder**, **desenvolvedor** e **verificador**.

- O **líder** tem como funções coordenar a equipa, fazendo o trabalho preparatório de cada etapa, distribuindo as responsabilidades pelos restantes membros e acompanhando a sua evolução. O líder é responsável por fazer as propostas preliminares de abordagem aos problemas, com detalhe suficiente para que cada membro possa avançar com o seu processo individual. Por exemplo, num diagrama de componentes, o líder deve explicitar as responsabilidades de cada componente e solicitar aos responsáveis por esse componente que as concretizem em termos de serviços por ele disponibilizados ou a ele necessários; compete ao líder também organizar as tarefas do verificador e usar os resultados delas para coadunar contributos iniciais que não estejam conformes ao necessário. O líder deve acompanhar a execução das tarefas pelos restantes membros, para deteção atempada de necessidades de afinação dos planos originais. Finalmente, o líder é o porta-voz da equipa, apresentando nos fóruns os resultados parcelares da equipa e resumindo dúvidas surgidas nas discussões internas, sendo também quem entrega as atividades nos dispositivos "Trabalho" (mas não as restantes atividades formativas, que são feitas individualmente). O líder deverá ter componentes de desenvolvimento a seu cargo, mas com dimensão e volumes de trabalho inferiores aos dos desenvolvedores, para não ser sobrecarregado face ao esforço de coordenação. Por ex., o líder poderá ser responsável por prototipar os métodos necessários e ligar métodos entre componentes, em vez de os implementar internamente; ou ser responsável pelo desenvolvimento de um componente menor.

- O **desenvolvedor** é um papel exercido por um a três estagiários da equipa (conforme a equipa tenha 3 a 5 integrantes). Os desenvolvedores são responsáveis por executar as tarefas de desenvolvimento de trabalhos previamente delineadas pelo líder e fazer a sua manutenção. Usando o exemplo acima, se o líder fizer os protótipos de métodos e os ligar entre componentes, compete ao desenvolvedor assegurar a funcionalidade interna desses métodos.

- O **verificador** deve verificar se as tarefas realizadas pelos desenvolvedores cumprem os requisitos previstos. Para tal, recebe do líder uma planificação e organização prévia dessa verificação e enquanto aguarda pelos desenvolvedores deve preparar o método e técnica necessários para as verificar. Pode e deve fazer verificações parcelares, não apenas aguardar pela conclusão do desenvolvimento. O verificador deverá ter componentes de desenvolvimento a seu cargo, mas com dimensão e volumes de trabalho inferiores aos dos desenvolvedores, para não ser sobrecarregado face ao esforço de verificação. Por ex., o líder poderá determinar que o verificador analisa se todas as atividades de input estão a ocorrer no componente View (ou Controller, conforme o estilo seguido) e o verificador analisa o projeto concreto para identificar onde e como deve procurar essas atividades de input; depois pode prosseguir com o desenvolvimento de um componente menor.

O tamanho ideal para a equipa é ter cinco membros, um líder, três desenvolvedores e um verificador. Para ilustrar a dinâmica de uma equipa atente à seguinte situação:

> *Numa atividade de decomposição de um projeto em componentes, o líder coordena o debate sobre essa decomposição arquitetónica. Após obtenção de consenso, especifica a atividade, indicando os componentes arquitetónicos e os serviços (métodos) fornecidos e consumidos por cada componente, bem como os parâmetros a receber/fornecer para o efeito. Os desenvolvedores verificam como devem estruturar internamente os componentes que lhes foram atribuídos, cumprindo essas indicações, para que quando chegar a ocasião de programar seja viável implementar a decomposição efetuada. O líder indica ao verificador quais os aspetos a verificar para assegurar que a decomposição feita por si, cumpre o exigido na atividade e quais os aspetos a verificar na estrutura interna dos componentes para estarem em conformidade com o especificado. O verificador começa por preparar instrumentos de orientação do seu trabalho de verificação e da recolha de elementos durante a verificação, iniciando-a pela decomposição do líder e depois pelo trabalho parcelar dos desenvolvedores, à medida que evolui.*

As equipas devem ter primordialmente uma organização de trabalho assíncrona. Para isso é importante estabelecer uma metodologia de comunicação, que cumpra as seguintes recomendações:

- Utilizar um sistema que permita organizar as mensagens entre os membros em linhas de debate ou canais, como ocorre aqui nos fóruns do Moodle. Uma forma prática é solicitar ao professor a criação de um fórum na disciplina para a equipa ou utilizar canais em plataformas mais ágeis como o Discord, que permite guardar ficheiros associados aos canais. (Deverá ser dado acesso a esses canais à equipa docente).

- Não troquem ficheiros de código por e-mail em anexos: utilizar-se-á um repositório de código, pelo que devem indicar o local onde se encontram no repositório os ficheiros relevantes.

- Para aspetos mais urgentes, devem usar um canal de comunicação com recibos de receção e leitura (“read/seen receipts”) como o Facebook Messenger, o Telegram, o Signal, o WhatsApp ou outros. Desta forma, sabem a cada momento se os restantes membros da equipa já receberam ou não a vossa mensagem.
- Devem reservar eventuais momentos síncronos (Skype, Zoom, etc.) apenas para situações excecionais. Respeitem a necessidade de gestão individual do tempo em estágio na SimProgramming em regime de e-learning: os momentos síncronos são sorvedouros de tempo, se usados com regularidade.

5. Recursos

Bibliografia e outros recursos

Bibliografia Obrigatória:

- Guerreiro, Sérgio (2015). Introdução à Engenharia de Software. FCA: Lisboa.

Recursos de software:

- Visual Studio Community - para desenvolvimento
- ChatGPT, Microsoft Copilot, outros - para assistente de inteligência artificial
- PlantText - para desenhar diagramas

6. Utilização de ferramentas de inteligência artificial

Utilização de ferramentas de Inteligência Artificial

Nota geral: utilização ética e responsável

Este tópico do contrato de aprendizagem foca-se na forma de utilização nesta unidade curricular de ferramentas de inteligência artificial capazes de gerar conteúdo original e criativo. Como exemplos, refiram-se as interfaces conversacionais (ChatGPT, Microsoft Copilot, Google Gemini, entre outras) ou as de geração de imagens (Midjourney, DALL-E, etc.) música, voz, espaços 3D, entre muitas outras que surgiram já e mais que virão a surgir.

A sua utilização deve ser encarada como de apoio e não de coautoria, na linha do que foi definido pelas principais editoras de revistas científicas, como a Springer-Nature, ou repositórios como o arXiv.

Adotam-se nesta UC estes dois princípios:

> ***1.º - Estas ferramentas não podem ter estatuto de coautoria de trabalhos académicos, porque essa autoria implica responsabilização pelo rigor, autenticidade e princípios éticos empregues na execução do trabalho.*** *No quadro legal atual, as ferramentas de inteligência artificial não podem assumir essa responsabilidade. Se os trabalhos incorrerem em plágio, erros factuais, referências falsas ou inadequadas às afirmações que visam sustentar, anfibolias na redação, ou meramente discorrerem superficialmente face ao solicitado, todas essas deficiências são da responsabilidade do autor que o apresenta, não da ferramenta que eventualmente as tenha gerado.* ***A analogia é que os erros ortográficos e gramaticais também são da responsabilidade do autor que apresenta o texto, não do corretor ortográfico-gramatical do processador de texto.***
>
> ***2.º - A produção de textos, imagens e outros materiais com ferramentas de inteligência artificial deve ter por base apoiar a melhoria da qualidade dos contributos desenvolvidos.*** *Não deve visar defraudar o objetivo central de aprender em comunidade nesta turma virtual. Recorde-se que o tempo, esforço e dedicação de todos os que colaboram no debate, apreciação e avaliação dos contributos mútuos (estudantes e docentes) é limitado e precioso. Desta forma, não é apenas fraude quando se usa estas ferramentas para produzir materiais que não se domina realmente. É também inadequado quando são empregues para insuflar o trabalho com matérias dispersas, carentes de contributos para de facto fazer evoluir os seus argumentos técnico-científicos.* ***A analogia é com práticas já anteriormente inadequadas, como discorrer longamente, sem objetivo, apenas para dar ilusão de erudição ou demonstrar ter empregado tempo na leitura (em vez de refletir e trazer os contributos resultantes dessa reflexão).***

==========

Casos particulares e sugestões de utilização

Se empregues dentro dos princípios acima apresentados, este tipo de ferramentas tem um potencial profundamente transformador das práticas profissionais e sociais. Por este motivo, encorajam-se os alunos a que as explorem e desenvolvam quadros próprios de emprego eficaz e eficiente. Encare sempre a resposta dos modelos como um diálogo no seu próprio processo de reflexão.
Tenha particular cuidado com a falácia de confundir a forma (texto sem erros ortográfico-gramaticais, texto assertivo) com o conteúdo ("quem fala bem não quer dizer que fale acertadamente"). Um caso é o designado fenómeno da alucinação. Por exemplo, se solicitar a escrita de código pode ocorrer esse código contar classes que não existem, ou métodos que não existem; ou que existam, mas com parâmetros de tipos diferentes dos produzidos. Em texto corrido, esses fenómenos ocorrem, são arquitetonicamente inevitáveis, mas por vezes muito subtis.

1. Identificação de estrutura no seu próprio texto

Pode solicitar aos modelos que analisem a estrutura do seu texto, sintetizando argumentos e detetando fragilidades argumentativas. Desta forma, pode melhorá-lo e clarificá-lo. Mas note que a resposta pode não ser correta: por exemplo, a síntese pode não ser rigorosa, inverter o sentido, omitir aspetos essenciais ou ter outras deficiências.
Por exemplo, eis uma análise do primeiro dos princípios feita pela versão gratuita do ChatGPT (modelo GPT-3) e pela versão mais avançada também gratuita (Microsoft Copilot).
É uma ajuda dialogante, mas atenção às suas deficiências. Apontam-se algumas:
No ChatGPT 3 (de forma não exaustiva):
- Note-se como interpretou "implica responsabilização por" como "devido à sua falta de responsabilização": no texto original, está-se a apresentar um estado, uma condição; na interpretação, está-se a apresentar um ato, uma decisão ou forma de atuar.

- Note-se como copiou *ipsis verbis* longas partes do texto original, que numa análise humana teriam de ser parafraseadas ou citadas entre aspas.
- Note-se como transformou "no quadro legal atual (...) não podem" em "a legislação atual não permite": transformou uma ausência de permissão na existência de uma proibição.

No Microsoft Copilot (GPT-4) (igualmente de forma não exaustiva):
- Note-se que fundiu a segunda premissa ("não podem assumir essa responsabilidade") com a conclusão ("não podem ser coautoras") (aspeto que também ocorreu no GPT-3.5).
- Na estrutura argumentativa, omitiu a analogia final.
- Note-se como na análise da coerência lógica só considerou uma das premissas, omitindo a outra.
- Na sustentação das conclusões, limitou-se a reafirmá-las associadas às premissas, não explicitou que elementos dos argumentos faziam a ligação lógica.

Interpelação	**Resposta do ChatGPT (GPT-3, 22/fev/2023)**
Estes princípios (indicados a seguir a "===") de utilização de ferramentas de inteligência artificial generativa destinam-se a uma unidade curricular de metodologias de investigação. Por favor, identifica: - as premissas; - a estrutura argumentativa; - a coerência lógica dos argumentos; - a sustentação das conclusões face aos pontos anteriores. === Estas ferramentas não podem ter estatuto de coautoria de trabalhos académicos, porque essa autoria implica responsabilização pelo rigor, autenticidade e princípios éticos empregues na execução do trabalho. No quadro legal atual, as ferramentas de inteligência artificial não podem assumir essa responsabilidade. Se os trabalhos incorrerem em plágio, erros factuais, referências falsas ou inadequadas às afirmações que visam sustentar, anfibolias na redação, ou meramente discorrerem superficialmente face ao solicitado, todas essas deficiências são da responsabilidade do autor que o apresenta, não da ferramenta que eventualmente as tenha gerado. A analogia é que os erros ortográficos e gramaticais também são da responsabilidade do autor que apresenta o texto, não do corretor ortográfico-gramatical do processador de texto.	Premissas: • As ferramentas de inteligência artificial generativa não podem ter estatuto de coautoria de trabalhos académicos devido à sua falta de responsabilização pelo rigor, autenticidade e princípios éticos empregues na execução do trabalho. • A legislação atual não permite que as ferramentas de inteligência artificial assumam essa responsabilidade. • Qualquer deficiência no trabalho, como plágio, erros factuais, referências falsas, anfibolias na redação ou superficialidade, é da responsabilidade do autor que o apresenta e não da ferramenta que gerou o trabalho. • A responsabilidade pelos erros ortográficos e gramaticais é do autor que apresenta o texto, não do corretor ortográfico-gramatical do processador de texto. Estrutura argumentativa: O argumento defende que as ferramentas de inteligência artificial generativa não podem ter estatuto de coautoria de trabalhos académicos devido à sua falta de responsabilidade e que todas as deficiências no trabalho são da responsabilidade do autor que o apresenta, não da ferramenta que gerou o trabalho. A analogia é feita com os erros ortográficos e gramaticais, que são da responsabilidade do autor que apresenta o texto, não do corretor ortográfico-gramatical do processador de texto. Coerência lógica dos argumentos: O argumento apresentado é coerente e lógico. As premissas são claras e bem definidas, e a analogia utilizada é apropriada para ilustrar a falta de responsabilidade das ferramentas de inteligência artificial generativa. O argumento é consistente com a legislação atual e com as práticas comuns de autoria académica. Sustentação das conclusões face aos pontos anteriores: As conclusões são bem sustentadas pelas premissas e argumentos apresentados. O argumento mostra que as ferramentas de inteligência artificial generativa não podem ter estatuto de coautoria de trabalhos académicos e que as deficiências no trabalho são da responsabilidade do autor que o apresenta. A analogia utilizada é convincente e ajuda a reforçar o argumento. As conclusões são consistentes com a legislação atual e com as práticas comuns de autoria académica.
	Microsoft Copilot (GPT-4, 28/fev/2024)

	Premissas: 1. A autoria de trabalhos académicos implica responsabilidade pelo rigor, autenticidade e princípios éticos. 2. As ferramentas de inteligência artificial generativa não podem ser coautoras de trabalhos académicos porque não podem ser responsabilizadas legalmente. **Estrutura Argumentativa:** O argumento é estruturado da seguinte forma: • **Afirmação inicial:** As ferramentas de IA não podem ter estatuto de coautoria. • **Justificação:** A coautoria implica responsabilidade legal que as ferramentas de IA não podem assumir. • **Exemplificação:** Se houver plágio ou erros, a responsabilidade recai sobre o autor humano, não sobre a ferramenta de IA. **Coerência Lógica dos Argumentos:** Os argumentos apresentados são logicamente coerentes. A responsabilidade legal é uma condição necessária para a coautoria, e como as ferramentas de IA não podem satisfazer essa condição, elas não podem ser consideradas coautoras. **Sustentação das Conclusões:** As conclusões são sustentadas pelos pontos anteriores, pois baseiam-se na premissa de que a responsabilidade legal é essencial para a autoria e, dado que as ferramentas de IA não podem ser legalmente responsáveis, elas não podem ser coautoras. A analogia final reforça o argumento ao comparar as ferramentas de IA com corretores ortográficos, que também não são responsáveis pelos erros do autor. Esta comparação ajuda a esclarecer que a ferramenta é apenas um meio para um fim, e não o autor final do conteúdo.

2. Sintetizar textos para apoiar a sua reflexão

Pode solicitar aos modelos de linguagem que extraiam as ideias principais de um texto ou do seu próprio texto, como apoio à sua reflexão.
Note que o modelo pode equivocar-se nesse processo, mas pode até encarar esses equívocos, em diálogo com o modelo, como forma de tentar explicitar melhor as suas ideias efetivas, ou detetar ambiguidades, necessidades de clarificação ou outras deficiências.
Por exemplo, eis um diálogo sobre o segundo dos princípios. Como exemplo de falha, realçaram-se a amarelo as partes que são cópia direta do texto original.
Repare que apesar de se ter pedido que essas cópias fossem evitadas ou incluídas entre aspas, substituídas por paráfrases, isso não ocorreu.

Pedido	**Resposta do ChatGPT (GPT-3, 22/fev/2023)**

Sintetiza estes princípios (indicados a seguir a "===") de utilização de ferramentas de inteligência artificial generativa numa unidade curricular de métodos de investigação. Extrai as ideias principais, como tópicos. Não copies texto diretamente do original, parafraseia. Se tiveres de usar texto do original, coloca-o citado, ou seja, entre aspas. === A produção de textos, imagens e outros materiais com ferramentas de inteligência artificial deve ter por base apoiar a melhoria da qualidade dos contributos desenvolvidos. Não deve visar defraudar o objetivo central de aprender em comunidade nesta turma virtual. Recorde-se que o tempo, esforço e dedicação de todos os que colaboram no debate, apreciação e avaliação dos contributos mútuos (estudantes e docentes) é limitado e precioso. Desta forma, não é apenas fraude quando se usa estas ferramentas para produzir materiais que não se domina realmente. É também inadequado quando são empregues para insuflar o trabalho com matérias dispersas, carentes de contributos para de facto fazer evoluir os seus argumentos científicos. A analogia é com práticas já anteriormente inadequadas, como discorrer longamente, sem objetivo, apenas para dar ilusão de erudição ou demonstrar ter empregado tempo na leitura (em vez de refletir e trazer os contributos resultantes dessa reflexão).	Aqui estão as ideias principais sintetizadas em forma de tópicos: • A produção de textos, imagens e outros materiais com ferramentas de inteligência artificial deve ter por base apoiar a melhoria da qualidade dos contributos desenvolvidos. • Não se deve usar essas ferramentas para defraudar o objetivo central de aprender em comunidade na turma virtual. • O tempo, esforço e dedicação de todos que colaboram no debate, apreciação e avaliação dos contributos mútuos (estudantes e docentes) é limitado e precioso. • É inadequado e considerado fraude quando se usa ferramentas de inteligência artificial para produzir materiais que não se domina realmente. • É também inadequado quando são empregues para encher o trabalho com matérias dispersas, carentes de contributos para de facto fazer evoluir os seus argumentos científicos. • Há uma analogia com práticas já anteriormente inadequadas, como discorrer longamente, sem objetivo, apenas para dar ilusão de erudição ou demonstrar ter empregado tempo na leitura. Em vez disso, é necessário refletir e trazer os contributos resultantes dessa reflexão.
Faz a síntese em menos de 50 palavras	O uso de ferramentas de inteligência artificial generativa em uma unidade curricular deve apoiar a qualidade dos contributos e não visar defraudar a comunidade virtual. É inadequado usar as ferramentas para produzir materiais que não se domina e para insuflar o trabalho com matérias dispersas.

3. Estruturar texto transcrito da oralidade

Pode gravar-se a falar, transcrever automaticamente com ferramentas como a transcrição do Word ou dos dispositivos móveis, e depois pedir ao ChatGPT para organizar o discurso sem redundâncias nem repetições, sem hesitações da oralidade, ordenando os argumentos de forma lógica e criando títulos e secções face à evolução dos conceitos.
Note que o modelo pode equivocar-se nesse processo, é sempre necessário encarar o processo de diálogo como apoio à sua reflexão, não como processo infalível.
Por exemplo, eis um diálogo sobre como encerrar estes exemplos e encorajar os alunos a procurar mais.
A sequência dos títulos indica o tema geral e as várias etapas do argumento. Contudo, o mero pedido de inclusão de títulos levou à alteração do argumento. Abandonando, por exemplo as referências iniciais de enquadramento e a antevisão da produção futura de mais materiais. Ou seja, podemos ter alterações de conteúdo quando pedimos apenas alterações de forma.

Deixamos ao seu critério, como atividade final, debater (no fórum de dúvidas geral) outras debilidades que detete neste exemplo final.

Pedido	**Resposta do ChatGPT (GPT-3, 22/fev/2023)**
Organiza este discurso (indicado a seguir a "==="), que foi gravado de uma pessoa a falar e transcrito automaticamente, para que fique sem redundâncias nem repetições, sem hesitações da oralidade. Ordena os argumentos de forma lógica e cria títulos e secções face à evolução dos conceitos. Emprega um tom semi-formal de professor universitário em Portugal, mas não demasiado distante do estilo do orador original. A extensão deve ser semelhante ao original. Este texto visa encerrar uma página onde se falou do ChatGPT e se deram princípios orientadores à sua utilização, pelo que deve ter teor de conclusão. === Ora bem caros estudantes podemos estar aqui a dar mais exemplos e no futuro se calhar até vamos criar vários recursos com exemplos sobre isto à medida que todos tenhamos mais experiência sobre isto mas para já o objetivo está atingido e para que este espaço não fique demasiado longo quero-vos dizer mais o que quero é que que possam inspirar-se e possam explorar esta ferramenta para serem todos mais eficientes e produtivos e não que se percam em tentativas soltas defraudar colegas defraudar professores ou esquivarem-se ao trabalho nada disso queremos é que em breve sejam mais capazes vamos todos para a frente.	Caros estudantes, atingimos o objetivo proposto e queria deixar-vos com algumas considerações finais sobre a utilização do ChatGPT. Em primeiro lugar, podemos continuar a explorar esta ferramenta e criar recursos com exemplos à medida que ganhamos mais experiência. Contudo, é importante lembrar que o nosso objetivo é utilizar o ChatGPT para sermos mais eficientes e produtivos, e não para defraudar colegas ou professores ou esquivar-nos ao trabalho. O que pretendemos é que possam inspirar-se e utilizar esta ferramenta de forma adequada, evitando tentativas soltas e carentes de contributo para o progresso dos seus argumentos científicos. Este espaço não deve ser usado para produzir materiais que não se dominam ou para encher trabalho com matérias dispersas. Em vez disso, a nossa visão é que todos possam ser mais capazes e evoluir os seus conhecimentos de forma construtiva. Utilizemos o ChatGPT de forma responsável, para ajudar-nos a produzir contributos de qualidade que enriqueçam o debate e a avaliação dos nossos trabalhos. Vamos todos para a frente.

Usa títulos e/ou subtítulos	**Utilização adequada do ChatGPT para aumentar a eficiência e produtividade** Caros estudantes, o ChatGPT é uma ferramenta poderosa para a produção de textos e outras tarefas que requerem o uso de linguagem natural. No entanto, é importante utilizá-la de forma adequada para aumentar a eficiência e produtividade. **Evitar tentativas soltas de defraudar colegas e professores** A utilização do ChatGPT deve ter por objetivo melhorar a qualidade dos trabalhos desenvolvidos e não defraudar colegas e professores. É essencial evitar tentativas soltas de enganar ou esquivar-se do trabalho. **Inspirar-se e explorar o ChatGPT de forma produtiva** Os estudantes devem inspirar-se e explorar o ChatGPT de forma produtiva para obterem o máximo benefício da ferramenta. É importante lembrar que, à medida que todos ganham mais experiência, é possível criar vários recursos e exemplos para a utilização adequada do ChatGPT. **Conclusão** Em suma, o objetivo deste espaço é atingido. Os estudantes devem utilizar o ChatGPT de forma adequada e produtiva para aumentar a sua eficiência e produtividade. Evitem defraudar colegas e professores e explorem a ferramenta para serem mais capazes e estarem prontos para enfrentar novos desafios no futuro.

7. Avaliação

Como vai ser a avaliação?

A avaliação assume o regime de avaliação contínua. Os estudantes que não possam seguir esta modalidade de avaliação podem optar pela realização de um Exame Final presencial.

A opção pelo regime de avaliação (contínua ou final) será feita pelo estudante até ao final **da terceira semana letiva**, não podendo ser alterada no decurso do semestre.

Os estudantes que optem pela avaliação contínua disporão de um **Cartão de Aprendizagem**, pessoal, onde será creditada a avaliação que forem efetuando ao longo do semestre. O Cartão de Aprendizagem é um instrumento personalizado e cada estudante tem acesso apenas ao seu cartão.

Os estudantes que optem pela realização de um Exame Final terão acesso a todas as orientações dadas na plataforma e às Atividades disponibilizadas ao longo do percurso de aprendizagem, mas não serão avaliados no regime de avaliação contínua - *E-fólios e E-fólio Global*. Estes estarão disponíveis apenas para os estudantes que optem pela modalidade de avaliação contínua.

Nota sobre os e-fólios:

- Tal como se refere na secção sobre Metodologia, neste PUC, adota-se uma abordagem por portfólio.
- Isto significa que NÃO HÁ UM ENUNCIADO NOVO PARA CADA E-FÓLIO: os alunos devem ir guardando, num portfólio pessoal, as atividades desenvolvidas ao longo das várias semanas de atividades, dentro da sua flexibilidade temporal e gestão pessoal do tempo e do esforço. Será o conjunto de resultados dessas atividades o elemento de avaliação na semana "e-fólio".
- Há atividades desenvolvidas a título individual (por ex., debate em fórum, realização de testes formativos) e atividades desenvolvidas em equipa (por ex. entregas intermédias do projeto).
- Como o projeto transversal é desenvolvido em equipa, constitui apenas uma entrega de portfólio para toda a equipa, mas deve incluir elementos específicos que identificam os contributos individuais (por ex., partes da estruturação, partes do código, testes específicos, etc.).

7.1. Cartão de Aprendizagem

Avaliação Contínua

Os estudantes que optarem pela avaliação contínua creditam no seu Cartão de Aprendizagem – CAP – as classificações que obtiveram com a realização de vários trabalhos ao longo do semestre e a classificação obtida numa prova presencial, no final do semestre.

Ao longo do semestre ser-lhes-á proposto que elaborem pequenos trabalhos que manterão num portfólio digital. Em datas específicas, os protótipos serão avaliados quantitativamente, momentos esses designados ***e-fólios***. A avaliação dos e-fólios poderá levar à creditação no seu cartão (CAP) de um máximo de 8 valores.

À classificação obtida nos ***e-fólios*** juntam-se os valores obtidos na prova final, que seguirá o regulamento de avaliação em vigor (***e-fólio global***) tendo esta a duração de 90min. Esta prova terá a cotação global de 12 valores.

O e-fólio global colocará questões que se apoiarão ou desenvolverão os trabalhos desenvolvidos nos e-fólios, não apenas (mas também) questões independentes.

No sistema de avaliação contínua, a classificação final na unidade curricular corresponde ao somatório das classificações obtidas em cada *e-fólio* e no *e-fólio global*.

Um ***e-fólio*** é um portfólio digital de trabalhos realizados nas semanas precedentes. Não é produzido numa semana específica: vai sendo desenvolvido pelos alunos, dentro da sua gestão do tempo e do esforço, ao longo do semestre. As "semanas de e-fólio" marcam apenas os momentos em que o professor irá avaliar as atividades incluídas no portfólio, pelo que terão de estar concluídas até esse momento.

O ***e-fólio global*** consiste num documento escrito a realizar nas condições previstas pelos regulamentos de avaliação da Universidade Aberta, que complementa os portfólios realizados eletronicamente e avaliados nas semanas "e-*fólios*".

As indicações para a realização quer das atividades para os portfólios ("e-*fólios*"), quer do *e-fólio global* serão fornecidas no decurso das semanas de atividades letivas. As cotações respetivas constam do plano de trabalhos, neste PUC.

A aprovação na Unidade Curricular implica que o estudante obtenha um mínimo de 3,5 valores (em 8) no conjunto dos *E-fólios* e um mínimo de 5,5 valores (em 12) no *E-fólio Global*.

7.2. Calendário de avaliação contínua

Calendário de avaliação contínua

Tópico	Conteúdo	Datas	Momentos dos e-fólios
1	Apresentações, objetivos, avaliação, instalação de software	4 a 17 de março	**Atividades feitas nas primeiras semanas (constituem o e-fólio 1): avaliadas a partir de 8 de abril**
2	Design de software: princípios Estilos arquitetónicos (MVC)	18 de março a 7 de abril	
3	Separação de interesses: conceito e abordagens Implementação de MVC com eventos	8 a 21 de abril	**Atividades feitos nas semanas intermédias (constituem o e-fólio 2): avaliadas a partir de 6 de maio**
4	Tipos de testes Tratamento de erros (exceções em MVC)	22 de abril a 5 de maio	
5	Separação de interface e implementação: conceito e abordagens; MVC com interfaces em C#	6 a 19 de maio	**Trabalhos feitos nas semanas finais (constituem o e-fólio 3): avaliados a partir de 3 de junho**
6	Melhorias e correções no projeto.	20 de maio a 2 de junho	

7.3. Exame

Opção pelo Exame

O Exame Final traduz-se numa única prova escrita realizada presencialmente e classificada numa escala de 0 (zero) a 20 (vinte) valores.

Esta prova tem um caráter somativo, reportando-se à totalidade do conteúdo abordado na unidade curricular e é realizada no final do semestre letivo, tendo a duração de 2h e 30min.

O estudante que optar pela modalidade de exame final, terá de obter neste um mínimo de 10 valores.

8. Plano de Trabalho

Calendário e plano de trabalho

Este *Plano* apresenta a previsão da distribuição temporal das várias **Temáticas** de estudo, das atividades e respetivas orientações de trabalho, de modo que possa planear, organizar e desenvolver o seu estudo. Esta informação é complementada por orientações que deverá consultar com regularidade nesta *Sala de Aula Virtual*.

semana	2.ª.feira	Tópico	Conteúdo téorico	Atividades	
1	4 de março de 2024	1	Apresentações, objetivos, avaliação, instalação de software	Apresentar-se; debater a avaliação; instalar o software; optar entre avaliação por exame ou contínua.	Estas atividades constituem o e-fólio 1 (são avaliadas a partir de 8 de abril de 2024)
				Ler a declaração de consentimento informado para participação em investigação (e aceitar ou recusar). Preencher o questionário de caracterização de perfil de aluno relativo às estratégias de autorregulação e corregulação das aprendizagens (só se tiver aceitado participar na investigação)	
2	11 de março de 2024			Escolher a equipa e o líder da equipa	
				Estudo teórico dos materiais recomendados e realização das atividades formativas.	
				Preencher a ficha de reflexão sobre a evolução quinzenal.	
3	18 de março de 2024	2	Design de software: princípios; Estilos arquitetónicos (MVC)	Escolher a API	
				Propor o tema de projeto a desenvolver	
				Estudo teórico dos materiais recomendados e realização das atividades formativas.	
Pausa letiva	25 de março de 2024				
4	1 de abril de 2024			Análise e estruturação em componentes MVC (Model-View-Controller) da solução para o projeto a desenvolver.	
				Preencher a ficha de reflexão sobre a evolução quinzenal.	
5	8 de abril de 2024	3	Separação de interesses: conceito e abordagens; Implementação de MVC com eventos	Definir as funções de cada elemento da equipa	Estas atividades constituem o e-fólio 2 (são avaliadas a partir de 6 de maio de 2024)
				Estudo teórico dos materiais recomendados e realização das atividades formativas.	
				Implementar a comunicação entre os componentes da arquitetura, com eventos e delegados	
6	15 de abril de 2024			Debater com a equipa a abordagem seguida	
				Preencher a ficha de reflexão sobre a evolução quinzenal.	
7	22 de abril de 2024	4	Tipos de testes; tratamento de erros; exceções em MVC	Estudo teórico dos materiais recomendados e realização das atividades formativas.	
				Conceber testes para o projeto.	
				Implementar tratamento de erros com recurso a exceções, respeitando a arquitetura MVC.	
8	29 de abril de 2024			Debater com a equipa a abordagem seguida	
				Relatório sobre o projeto até ao momento.	
				Preencher a ficha de reflexão sobre a evolução quinzenal.	
9	6 de maio de 2024	5	Separação de interface e implementação: conceito e abordagens; MVC com interfaces em C#	Estudo teórico dos materiais recomendados.	Estas atividades constituem o e-fólio 3 (são avaliadas a partir de 3 de junho de 2024)
				Conceber interfaces de código para acoplamento mais fraco e mais independência entre componentes do projeto	
				Implementar interfaces de código no projeto	
				Debater com a equipa a abordagem seguida	
10	13 de maio de 2024			Relatório sobre o projeto até ao momento.	
				Preencher a ficha de reflexão sobre a evolução quinzenal.	
11	20 de maio de 2024	6	Fundamentos da construção de software; conclusão do projeto	Corrigir e melhorar o projeto	
				Debater com a equipa a abordagem seguida	
12	27 de maio de 2024			Relatório final sobre o projeto	
				Preencher a ficha de reflexão sobre a evolução quinzenal.	
				Preencher as fichas de autoavaliação e heteroavaliação.	
Fim	2 de junho de 2024				
			e-fólio global ou exame: 17/6/2024 (manhã). Recurso: 10/9/2024 (manhã)		

Cotações	
Itens	**Cotação**
As Regras Fórum	0,05
Precisa de ajuda? Fórum	0,05
Ritual de iniciação dos estagiários Fórum	0,05
Autoconhecimento: Sou Programador?! Teste	0,05
Handbook da SimProgramming Teste	0,05
Post-mortems de dois programas Teste	0,05
Impactos da Engenharia de Software Teste	0,05
Atacar o desenvolvimento de software em equipa Teste	0,05
Responsabilidades e funções das equipas Teste	0,05
Preparação da jornada Teste	0,05
Aventuras em códigos legados Teste	0,05
Ajuda com um projeto ensarilhado Trabalho	0,2
A minha primeira quinzena! Teste	0,05
Nova fase, precisas de ajuda? Fórum	0,05
Ferramentas ... Qual escolher? Fórum	0,05
À descoberta do que sei sobre uma API Teste	0,05
Em busca dos fundamentos Teste	0,05
Compreensão de um programa Livro	0
Mapeamento de E/S e MVC Teste	0,05
Entradas vs. Saídas... já as consigo distinguir? Teste	0,05
Decompor e arquitetar Teste	0,05
O fluxo da arquitetura Fórum	0,25
Reflexões sobre a decomposição arquitetónica. Teste	0,05
Modelação da arquitetura Trabalho	0,5

Encerramento do 1.º mês na SimProgramming Teste	0,05
Total do e-fólio 1	**2**
Ajuda nos entraves arquitetónicos Fórum	0,05
Código Inicial do projeto Formas Aleatórias Recurso	0
Código Inicial do projeto do Quarto Chines Recurso	0
Ferramentas para a harmonia arquitetónica Teste	0,05
Compreensão das dinâmicas arquitetónicas Livro	0
Força e debilidade da arquitetura Teste	0,05
Identificação de forças e debilidades arquitetónicas Teste	0,05
Reflexões sobre referências e acoplamentos Fórum	0,05
Indagações sobre as interações dos componentes Teste	0,05
Mudanças no fluxo da arquitetura Trabalho	0,3
Dimensões da qualidade ... O que já sei? Teste	0,05
Implementação da aplicação demonstradora Fórum	0,15
Código da aplicação demonstradora da Ada e da Fezada com eventos e delegados Recurso	0
Código da aplicação demonstradora do Meiabola e do Patavinas com eventos e delegados Recurso	0
Auto-organização para a implementação Trabalho	0,5
Instruções para criação de Git no Bitbucket Página	0,05
A quantas ando na implementação? Teste	0,05
Apagar incêndios da implementação Teste	0,05
Um cafezinho para pensar nas falhas e faltas Teste	0,05
Teste, teste, teste ... Porquê? Teste	0,05
Mapear os testes Trabalho	0,3
Minuta do plano de testes Recurso	0
Controlar os erros Fórum	0,1
Habemus aplicação? Trabalho	1
Código da aplicação demonstradora Formas Aleatórias, com exceções em MVC Recurso	0
Código da aplicação demonstradora Tradutor do Quarto Chinês, com exceções em MVC Recurso	0
Como estou até este ponto? Teste	0,05
Total do e-fólio 2	**3**
Evolução dos componentes da aplicação Fórum	0,05
Lista de dependências informativas: Tradutor do Quarto Chinês Recurso	0
Lista de dependências informativas: Gerador de Formas Aleatórias Recurso	0
Modelo para a lista de dependências da sala de desenvolvimento C40 Recurso	0
Reflexões sobre o processo de implementação Teste	0,05
Análise das dependências Trabalho	0,15
Relatório de Dependências: Tradutor do Quarto Chinês Recurso	0
Relatório de Dependências: Gerador de Formas Aleatórias Recurso	0
Modelo para o relatório de dependências Recurso	0
Máscaras para componentes - as interfaces Teste	0,05
Reflexões sobre a independência de componentes Fórum	0,05
Desenvolvimento de interfaces Fórum	0,05
Mãos à massa com as interfaces Trabalho	0,75
Tradutor do Quarto Chinês Recurso	0
Gerador de Formas Aleatórias Recurso	0
Como estou a utilizar as interfaces? Teste	0,05
Está quase! Reflexões na reta final da implementação Teste	0,05
Apresentação da aplicação demonstradora Fórum	0,4
Relatório final da aplicação demonstradora Trabalho	1,2

Linha de chegada! Como foi o percurso na SimProgramming? Teste	0,15
Ficha de pós-caracterização do seu perfil de estratégias de autorregulação e corregulação das aprendizagens	0
Total do e-fólio 3	**3**