

Modelo de Relatório Detalhado da Experiência com LLM

1. Identificação do Grupo

Nome do Aluno 1: Pedro Cardoso

Nome do Aluno 2: Tomás Pica

Nome da Aplicação Desenvolvida: Sistema de votação

Data de Início: 29/05/26 Data de Conclusão: 12/06/26

Durante cada fase, os alunos devem preencher a tabela abaixo para cada iteração. Cada vez que o LLM encontra uma limitação ou erro, os alunos devem trocar de papel e registrar os acontecimentos.

Link do projeto no github: <https://github.com/DRSulAGen-202425-students/projeto-sistema-votacao>

2. Técnicas Aplicadas

Aplicação das Técnicas

Técnica: Decomposição e Modelação

Como foi aplicada:

A técnica de decomposição foi aplicada através da formulação de um prompt específico para cada requisito, permitindo uma abordagem modular e focada na resolução individual de cada parte do problema. Esta modelação incremental facilitou a obtenção de respostas mais precisas e adaptadas a cada componente do sistema, sem ocorrência de erros.

Técnica: Test-Driven Development

Como foi aplicada:

Antes da implementação final de várias classes e funcionalidades, foram definidos testes unitários que descreviam o comportamento esperado dos métodos. Esta abordagem orientou o desenvolvimento, assegurando que cada função fosse construída com base nos requisitos funcionais testáveis desde o início.

Observações:

O uso do TDD permite uma maior confiança na estabilidade do sistema ao longo das alterações. Além disso, facilita a deteção precoce de erros e a melhoria contínua da estrutura do código, conduzindo a uma cobertura total dos testes.

Técnica: Testes com JUnit

Como foi aplicada:

Foi utilizada a framework JUnit para desenvolver testes unitários para todas as classes principais do sistema, incluindo classes de modelo, serviços, repositórios e componentes de segurança. Cada teste foi desenvolvido com assertivas claras para verificar o comportamento esperado em diferentes cenários.

Observações:

Os testes permitiram validar corretamente a lógica do sistema, garantindo a robustez da aplicação. A cobertura de código alcançou 100% nas classes testadas, com casos abrangendo tanto os fluxos principais como exceções e estados inválidos.

Técnica: Code Review

Como foi aplicada: A técnica de code review foi aplicada após a geração de código pelo modelo. Cada trecho de código foi analisado manualmente para verificar a sua correção, legibilidade e conformidade com as boas práticas de programação. Foram feitas revisões iterativas, corrigindo eventuais erros, melhorando a estrutura do código e otimizando a sua eficiência.

Observações: Durante o processo de code review, identificaram-se pequenas inconsistências tais como nomes de variáveis errados.

3. Desenvolvimento da Aplicação

Tabela de Iterações

#	Nome Aluno	Funcionalidade Tentada	Resultado do LLM (Sucesso/Falha)	'LLM Avançado' (Sim/Não)	*Solução Manual ou Reformulação do Prompt? (M/R)	Observações (prompt escrito)
1	Pedro Cardoso	Introdução do trabalho e gerar o código para o requisito rf01	Sucesso	Não		"Estou a desenvolver uma aplicacao em Java com o objetivo de Criar um sistema de votação eletrónica que permita a eleitores registados votar em candidatos numa eleição, de forma segura, simples e verificável. Com os seguintes perfis Administrador: Gere candidatos, eleitores e visualiza resultados. Eleitor: Vota e visualiza confirmação do seu voto. estes sao os requesitos funcionais Requisitos Funcionais • RF01 – Autenticação de Utilizadores: O sistema deve permitir login de administradores e eleitores com credenciais únicas.

Universidade Lusófona de Humanidades e Tecnologias
Desenvolvimento Responsável de Software Baseado em IA Generativa
2024/25
Bruno Cipriano e Daniel Silveira
V1.0.0

						<ul style="list-style-type: none"> • RF02 – Gestão de Candidatos (Administrador): Inserir, editar e remover candidatos antes do início da votação. • RF03 – Gestão de Eleitores (Administrador): Inserir, editar e remover eleitores registados. • RF04 – Início e Encerramento da Votação (Administrador): Apenas o administrador pode iniciar e encerrar a votação. • RF05 – Processo de Votação (Eleitor): O eleitor pode votar apenas uma vez e de forma secreta. • RF06 – Apuramento de Resultados (Administrador): O sistema mostra total de votos, percentagens e o vencedor. • RF07 – Prevenção de Voto Duplo: O sistema deve garantir que cada eleitor vote apenas uma vez.
2	Pedro Cardoso	Gerar o código para o requisito rf02	Sucesso	Não		vamos começar pelo o rf01" "vamos agora para o rfo02"
3	Pedro Cardoso	Estruturar o projeto em várias pastas	Sucesso	Não		"estrutura o projeto em várias pastas "
4	Pedro Cardoso	Gerar o código para o requisito rf03	Sucesso	Não		"vamos agora para o rf03"
5	Pedro Cardoso	Gerar o código para o requisito rf04	Sucesso	Não		"vamos agora para o RF04"

Universidade Lusófona de Humanidades e Tecnologias
Desenvolvimento Responsável de Software Baseado em IA Generativa
2024/25
Bruno Cipriano e Daniel Silveira
V1.0.0

6	Pedro Cardoso	Gerar o código para o requisito rf05	Sucesso	Não		"vamos agora para o rf05"
7	Pedro Cardoso	Gerar o código para o requisito rf06	Sucesso	Não		"vamos agora para o rf06"
8	Pedro Cardoso	Gerar o código para o requisito rf07	Sucesso	Não		"vamos agora para o rf07"
9	Tomás Pica	Introdução dos requisitos não funcionais e gerar o código para o requisito rnf01	Sucesso	Não		<p>"agora temos requisitos não funcionais"</p> <p>Requisitos Não Funcionais</p> <ul style="list-style-type: none"> • RNF01 – Segurança: As credenciais devem ser protegidas e os votos registados de forma anónima. • RNF02 – Usabilidade: A interface deve ser simples e clara. • RNF03 – Persistência: Os dados devem ser armazenados numa base de dados. • RNF04 – Fiabilidade: Garantir integridade dos dados mesmo com falhas. • RNF05 – Portabilidade: A aplicação deve funcionar em qualquer sistema com Java instalado. <p>vamos começar pelo rnf01, com a mesma personalidade e tipo de resposta que dás anteriormente, indicando os packages e sem apagar código se possível, incrementa o que for necessário. mostra as alterações do ficheiro inteiro!"</p>

Universidade Lusófona de Humanidades e Tecnologias
Desenvolvimento Responsável de Software Baseado em IA Generativa
2024/25
Bruno Cipriano e Daniel Silveira
V1.0.0

10	Tomás Pica	Explicar o que já foi feito de modo a ter o código restante	Sucesso	Não		"Implementei o security e o User. nos ficheiros originais, o AuthService só usa UserRepository e o Vote usa electorUsername e candidateId"
11	Tomás Pica	Pedir que verifique as classes restantes de modo a descobrir 'buracos' e de seguida implementar uma nova funcionalidade	Sucesso	Não		"verifica se existe código em falta nas restantes classes e continua a implementar os votos anónimos"
12	Tomás Pica	Atualizar um método de uma determinada subclasse	Sucesso	Não		"atualiza o setPassword do Elector que é filho de User. o security tem o hashPassword"
13	Tomás Pica	Implementar a nova funcionalidade	Falha	Não	Reformulação	"faz agora os votos anónimos"
14	Tomás Pica	Explicar que algo está errado e pedir para corrigir	Sucesso	Não		"o votingService está errado e nós estamos a testar tudo no Main. tens todo o código do programa, são os requisitos funcionais"
15	Tomás Pica	Procurar alternativas melhores, mais simples, organizadas e corretas nas classes	Sucesso	Não		"o Elector já tem um parâmetro hasVoted e um método"
16	Tomás Pica	Explicar o que deve usar no código e o que não remover	Sucesso	Não		"O VotingService tem CandidadeRepository, voteRepository e votingSessio"
17	Tomás Pica	Dar contexto mais recente de modo a conseguir visualizar uma solução	Sucesso	Não		" Como posso continuar daqui? Eu tenho de ver se o Elector votou e o voto tem de ser anónimo: *código VotingService*"
18	Tomás Pica	Pedir que verifique se está tudo correto	Sucesso	Não		"Verifica se está bem *várias classes*"
19	Tomás Pica	'Output' da solução anterior de modo a corrigir erros	Sucesso	Não		"No main dá credenciais inválidas e o Elector pode ser null."
20	Tomás Pica	Verificar o código novamente	Sucesso	Não		"Já não deu erros. verifica o rnf01 para ver se está tudo"

Universidade Lusófona de Humanidades e Tecnologias
Desenvolvimento Responsável de Software Baseado em IA Generativa
2024/25
Bruno Cipriano e Daniel Silveira
V1.0.0

21	Tomás Pica	Verificar se os requisitos NF estão implementados	Sucesso	Não		"verifica se os rnf's estão implementados"
22	Tomás Pica	Dar continuidade aos requisitos começando pelo rnf3	Sucesso	Não		"vamos agora para o rnf03 Modelo de Dados Tabela: Eleitor (id, nome, numero_eleitor, username, password_hash, votou) Tabela: Candidato (id, nome, partido, numero) Tabela: Voto (id, id_candidato, timestamp) faz de acordo com código que temos agora"
23	Tomás Pica	Reformular ou corrigir o código do rnf3	Falha	Não	Reformulação	"Implementei o util - que se passou a chamar database. Dá-me o código atualizado: *código databases*"
24	Tomás Pica	Implementar testes unitários junit	Sucesso	Não		"quero organizar os testes, dá-me os testes junit / unitários para todas as classes do programa de forma a ter cobertura 100%"
25	Tomás Pica	Gerar testes para todas as outras classes	Sucesso	Não		"Gera para tudo, para todas as classes models, service, repository (só falta o UserRepository) e security."
26	Tomás Pica	Dar código de classes que restam para dar mais contexto de modo a fazer os testes corretamente	Sucesso	Não		"Dá me novamente e corretos os testes para as classes VotingSession, VotingService, Vote, UserRepository e ResultService de acordo com o código: *código classes*"
27	Tomás Pica	Implementar testes das classes que restam	Sucesso	Não		"Dá me os testes para estas classes também: *código das últimas classes*"
28	Tomás Pica	Avisar que faltam testes para a classe User de modo a implementá-los	Sucesso	Não		"Falta um método no UserTests, 66% coverage"

4. Exploração de Código com LLM

Exploração de Código (copy paste)

Classe/Função	Pergunta feita ao LLM	Explicação dada pelo LLM	A explicação foi correta? Justificação
Vote Repository / Voting Service	"o Elector já tem um parâmetro hasVoted e um método"	"1.Manter o Vote anónimo (só candidateId). 2.Fazer a verificação e atualização de hasVoted diretamente em Elector. 3.Injetar o UserRepository (ou ElectorRepository) no VotingService para buscar e marcar o eleitor. "	Sim, pois estas decisões garantem que o sistema respeita o voto anónimo, assegurando que apenas o identificador do candidato é armazenado no repositório de votos, enquanto o estado de votação é controlado separadamente no objeto Elector.
Voting Service	"Como posso continuar daqui? Eu tenho de ver se o Elector votou e o voto tem de ser anónimo: *código VotingService*"	"1.Verificar se a votação está ativa 2.Validar que o utilizador é um Elector e que ainda não votou (usando o campo hasVoted do próprio Elector) 3.Validar existência do candidato 4.Registar o voto (anónimo) — adicionando um único Vote que só contém o candidateId 5.Marcar hasVoted = true no Elector"	Sim, estas etapas garantem que apenas eleitores válidos participam, evitando votos duplicados e mantendo o voto anónimo.

UserTest	"Falta um método no UserTests, 66% coverage"	"Boa observação. O método faltando no UserTest é provavelmente o abstrato getRole(), que está definido mas não diretamente testado no contexto da subclasse de teste."	Sim, só havia um método que não estava a ser testado, neste caso era o getRole().
----------	--	--	---

5. Reflexão Final

- Que vantagens e desvantagens identificaram no uso de LLM?

Durante a realização das tarefas, o uso de modelos de linguagem (LLM) revelou-se bastante eficaz. As principais vantagens identificadas foram:

- Rapidez na geração de código e textos explicativos;
- Facilidade de adaptação das respostas a diferentes requisitos e estilos;
- Ajuda na decomposição de problemas, com sugestões estruturadas e orientadas;
- Redução de tempo de desenvolvimento, especialmente em tarefas repetitivas ou padrão.

Desvantagens:

Neste caso específico, não foram identificadas desvantagens significativas. Todas as interações com o modelo forneceram respostas úteis e facilmente adaptáveis, não tendo sido necessário recorrer a soluções alternativas manuais complexos.

- Que tarefas foram simplificadas?

O uso do LLM simplificou a implementação das principais funcionalidades do sistema, como autenticação, gestão de dados e lógica de votação. Facilitou também a estruturação do código em Java e a escrita de métodos com base nos requisitos, reduzindo significativamente o tempo de desenvolvimento.

- Que tarefas continuam a requerer pensamento crítico?

Apesar do apoio do LLM, o planeamento do projeto, a definição clara dos requisitos, a validação da lógica e a garantia da segurança e integridade dos dados continuam a requerer análise e decisão crítica por parte do programador.

- Qual acham que será o papel do engenheiro de software num futuro com LLMs avançadas?

Num futuro com LLMs avançadas, o papel do engenheiro de software irá evoluir para se focar mais na identificação de problemas, supervisão da qualidade do código gerado, integração dos sistemas e tomada de decisões.

5. Anexo Exemplo

#	Nome Aluno	Funcionalidade Tentada	Resultado do LLM (Sucesso/Falha)	'LLM Avançado' (Sim/Não)	*Solução Manual ou Reformulação do Prompt? (M/R)	Observações (prompt escrito)
1	PMau	Classe Desenha_Menu	S	N		Faz funcao que desenha o menu.
2	PMau	Classe Chumba_Alunos	F	N	R	Cria algoritmo que chumba sempre 75% dos alunos
3	PMau	Classe Chumba_Alunos	F	N	R	Cria calculo que chumba sempre 75% dos alunos maus
4	PBom	Classe Chumba_Alunos	S	S		Cria calculo que chumba sempre 75% dos alunos maus
5	PBom	Classe Envia email da nota	F	N	R	Faz Funcao envia email para lista de alunos com a nota
6	PBom	Classe Envia email da nota	F	N	R	Faz Funcao envia email para lista de alunos com a nota

Universidade Lusófona de Humanidades e Tecnologias
Desenvolvimento Responsável de Software Baseado em IA Generativa
2024/25
Bruno Cipriano e Daniel Silveira
V1.0.0

6	PBom	Classe Envia email da nota	F	N	R	Faz Funcao envia email com conta google para lista de alunos com a nota
7	PBom	Classe Envia email da nota	F	S	M	Faz Funcao envia email com conta google para lista de alunos com a nota
8	PBom	Classe Envia email da nota	S	N	M	Após conversarmos o PBom a tentat ser LLM Avancado não estava conseguir percebe porque faltava dizer que era pra usar o servido emails pre implementado
9	PBom	Classe Lixa_Aluno	...			