Relatório Detalhado da Experiência com LLM

# 1. Identificação do Grupo

Nome do Aluno 1: Ana Beatriz Pires a22109141

Nome do Aluno 2: André Cunha a22400787

Nome da Aplicação Desenvolvida: Sistema de Votação Eletrónica

Data de Início: 29/05/2025. Data de Conclusão:03/06/2025

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 registar os acontecimentos.

Link do projeto no github: <https://github.com/DRSuIAGen-202425-students/projeto-andre_bia>

# 2. Técnicas Aplicadas

## Aplicação das Técnicas

* Técnica: Decomposição e Modelação

Como foi aplicada:

Começámos por enviar o ficheiro do guião com os requisitos da aplicação para o LLM e pedimos que ele fizesse a análise e a decomposição dos requisitos no documento em várias componentes. Após isso fizemos uma análise cuidada sobre a resposta do LLM para verificar se a decomposição e as Classes que o LLM sugeriu faziam sentido para o projeto.

Observações:

* Técnica: Test-Driven Development

Como foi aplicada:

Após verificarmos a decomposição pedimos ao LLM para elaborar testes em Java sobre as classes que este tinha encontrado, verificando sempre se os testes criados faziam sentido. Depois, solicitámos que o LLM começasse a escrever o código das classes tendo em conta os testes que tinha criado.

Numa primeira abordagem subestimamos o tamanho do projeto e percebemos que o LLM se tinha perdido um pouco nas funcionalidades dado à janela de contexto ser demasiado pequena, pelo que decidimos fazer *prompts* mais cirúrgicos para cada classe em vez de pedirmos a implementação de todas as classes de uma vez só.

Observações:

* Técnica: Testes com JUnit

Como foi aplicada:

Os Testes foram gerados pelo LLM com base nos requisitos.

Observações:

* Técnica: Code Review

Como foi aplicada:

Observações:

# 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 | André | Preparação dos prompts | S | N |  | Pedi ao LLM os prompts que me indicasse quais os prompts que tinha de lhe dar para que não se esquecesse de nada. |
| 2 | André | Classe Eleitor /Utilizador | S | N |  | Dá-me a classe Java Eleitor, que herda de Utilizador e tem o campo boolean votou. Inclui construtor, getters/setters e toString. |
| 3 | André | Classe Admnistrador | S | N |  | Dá-me a classe Java Administrador, que herda de Utilizador e representa o admin com permissões especiais |
| 4 | André | Classe Candidato | S | N |  | Dá-me a classe Java Candidato que representa um candidato e contém id, nome, partido, numeroVotos |
| 5 | André | Classe Votacao | S | N |  | Dá-me a classe Java Votacao que representa o estado atual da votação, ativa e datas |
| 6 | André | Classe Voto | S | N |  | Dá-me a classe Java Voto, que representa um registo de voto |
| 7 | André | Classe VotacaoService | S | N |  | Dá-me o código da classe VotacaoService com os métodos: iniciarVotacao(), encerrarVotacao(), votar (Eleitor, Candidato) e isVotacaoAtiva(). |
| 8 | André | Classe AutenticacaoService | S | N |  | Dá-me o código da classe AutenticacaoService com os métodos responsavel por validar as credenciais de login |
| 9 | André | Classe UtilizadorRepository | S | N |  | Dá-me a classe UtilizadorRepository que guarda utilizadores numa list, com métodos para login, adicionar utilizador, editar e apagar. |
| 10 | André | Classe GestaoCandidatosService | S | N |  | Dá-me o código Java GestaoCandidatosService com os métodos para insrir, editar e remover um cadidato |
| 11 | André | Classe CandidatoRepository | S | N |  | CandidatoRepository |
| 12 | André | Classe GestaoEleitoresService | S | N |  | Dá-me o código Java GestaoEleitoresService com métodos para inserir, editar e remover eleitores |
| 13 | André | Classe EleitorRepository | S | N |  | da-me o código completo de EleitorRepository |
| 14 | André | Classe ResultadosServices | S | N |  | Dá-me o código Java ResultadosServices com métodos para calcular totais, percentagens e vencedor |
| 15 | André | Classe Main | N | N | M | Dá-me uma classe Main com menu em consola que permita ao utilizador autenticar-se e, consoante o tipo, aceder às opções de administrador ou eleitor. **Foram necessários fazer algumas correções manuais aos nomes de alguns métodos que apesar da lógica estar implementada eram chamados no main com nome errado.** |
| 16 | André | Classe de Testes | N | N | R | Pedi para gerar os testes unitários com base nos requisitos e nas classes implementadas (listando cada um dos requisitos e classes). |
| 17 | André | Classe Testes | N | N | M | Enviei o excerto de código com erro gerado pelo LLM e pedi que o corrigisse**. Foi novamente necessário corrigir a chamada de métodos com nome errado** |
| 18 | André | Main (Orquestrador) | S | N |  | Pedi o código do Main detalhando o fluxo de execução do programa pretendido. **Código gerado continha muitos erros, maioritariamente com chamada de métodos que não existiam de todo.** |
| 19 | André | Classe GestaoCandidatoService | S | N |  | Pedi o Código da classe GestaoCandidatoService |
| 20 | André | Classe CandidatoRepository | S | N |  | Pedi o Código da classe CandidatoRepository |
| 21 | André | Classe UtilizadorRepository | S | N |  | Pedi o Código da classe UtilizadorRepository |
| 22 | André | Classe ResultadosService | S | N |  | Pedi o Código da classe ResultadosService |
| 23 | André | Classe EleitorRepository | S | N |  | Pedi o Código da classe EleitorRepository |
| 24 | André | Classe Testes | N | N | M | Enviei o Código de Testes ainda com alguns erros na chamada dos métodos e pedi que os corrigisse tendo em conta as alterações que fizemos ás classes nos prompts anteriores. **Foi novamente necessário corrigir a chamada de métodos com nome errado** |
| 25 | André | Classe UtilizadorRepository | S | N |  | Pedi que me desse o Código do método encontrarPorUsername() que estava a ser chamado nos testes |
| 26 | André | Classe Main | N | N | R | Pedi que me desse o código do Main sem misturar a responsabilidades de cada módulo |
| 27 |  |  |  |  |  | **Como a resposta foi pior que a solução anterior consideramos a utilização de uma conversa nova (novo chat)** |
| 28 | André | Classe Main | S | N |  | Mandei o codigo do Main e expliquei ao LLM o problema e o que queria resolver (respeitar a separação de responsabilidades) |
| 29 | André | Classe GestaoEleitoresService e GestaoCandidatosService | S | N |  | Pedi o novo Código da classe GestaoEleitoresService e GestaoCandidatosService |
| 30 | André | Classe Testes | N | N | R | Especifiquei os requisites no prompt e pedi que gerasse a classe de Testes JUnit |
| 31 | André | Classe Testes | S | N |  | Pedi que me desse o Código dos métodos em falta chamados na classe de Testes (especificando os métodos no prompt) |
| 32 | André | Classe Testes | S | N |  | Pedi que me desse o Código correto do método votar da classe VotacaoService que Segundo os testes deveria lançar uma exceção |
| 33 | André | Classe de Testes | S | N |  | Pedi que complementasse a classe de testes com testes de borda |
| 34 | Beatriz | Função encerrarVotação | S | N |  | Pedi que reformulasse a função encerrarVotacao para que “limpasse” o estado dos votantes para qualquer votação que acontecesse após terminar a 1ª |
| 35 | Beatriz | Implentação da Classe FileStorageUtil | S | N |  | Pedi que me implementasse a Classe FileStorageUtil que serve para guardar/carregar dados serializados em/a partir de ficheiros .dat |
| 36 | Beatriz | Implementação guardar dados | S | N |  | Pedi que o LLM transformasse todos os repositórios de forma a que guardasse os dados serializados em ficheiros .dat |

* Esta opção só acontece em caso falha.

# 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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 5. Reflexão Final

• Que vantagens e desvantagens identificaram no uso de LLM?

**Vantagens:**

- Eficaz quando o problema está bem definido e descrito com detalhe;

- Permite gerar código funcional mais rapidamente, acelerando o desenvolvimento;

**Desvantagens:**

- Inicialmente, é necessário algum tempo para compreender o contexto solução apresentada;  
 - A janela do contexto limitada dificulta a resolução de problemas mais complexos ou extensos;

- As soluções devem ser revistas com atenção, pois podem conter erros ou suposições incorretas;  
- Reformular o prompt nem sempre corrige falhas anteriores – o modelo pode manter informações de interações anteriores, mesmo quando já não são relevantes.

- Por vezes, o modelo gera soluções excessivamente complexas ou além do solicitado, o que pode dificultar a integração ou validação;

- Não é muito eficaz quando o problema é demasiado abstrato

• Que tarefas foram simplificadas?

- Organização do projeto em partes mais pequenas

- Correção de erros e a geração do código no geral.  
• Que tarefas continuam a requerer pensamento crítico?

- A estrutura do programa no geral e toda a validação da lógica do código gerado pelo LLM.  
• Qual acham que será o papel do engenheiro de software num futuro com LLMs avançadas?

- Acreditamos que o papel do engenheiro de software passará por usar LLMs para escrever boa parte do código nos projetos, mas que será sempre necessário pensamento crítico e conhecimento técnico profundo por parte do engenheiro de software e inputs ocasionais de código feito à mão para evitar bugs e falhas graves de segurança que podem comprometer os sistemas de forma grave

# 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 |
| 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 | … |  |  |  |