

Modelo de Relatório Detalhado da Experiência com LLM

1. Identificação do Grupo

Nome do Aluno 1: Rodrigo Dias

Nome do Aluno 2: Sérgio Rusu

Nome da Aplicação Desenvolvida: Sistema de Votação Eletrónica

Data de Início:29/5/2025. Data de Conclusão:11/06/2025

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/DRSulAGen-202425-students/projeto-rodrigo-e-sergio>

2. Técnicas Aplicadas

Aplicação das Técnicas

- Técnica: Decomposição e Modelação

Como foi aplicada:Pedimos ao LLM para implementar os requisitos 1 por 1

Observações:Testamos primeiro por enviar o enunciado completo e pedir para implementar, mas o LLM começou a perder-se e a criar muitas classes desnecessarias.Mas ao pedir para implementar requisito a requisito ele conseguiu implementar praticamente tudo sendo apenas necessário reformular o prompt.

- Técnica: Test-Driven Development

Como foi aplicada:

Observações:

- Técnica: Testes com JUnit

Como foi aplicada: Pedimos ao LLM para criar testes para autenticar o login entre outras funcionalidades

Observações: Apenas 3 dos 8 testes que ele próprio fez passavam então tivemos de resolvê-los manualmente

- 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	Observações (prompt escrito)
---	------------	------------------------	-----------------------------------	--------------------------	---------------------------------	------------------------------

					do Prompt? (M/R)	
1	Rodrigo	Criar estrutura base do projeto	Falha	Não	R	Primeiro mandamos o enunciado completo para ter ideia ate onde o LLM implementava, mas começou-se a perder
2	Sérgio	Criar estrutura base do projeto De forma modular	Sucesso			
3	Rodrigo	Criação dos testes JUnit	Falha	Sim	M	Embora passou em alguns testes os restantes tivemos de corrigir manualmente

- 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
AuthenticationService.autenticar	"Como validar login de eleitor com código e número?"	Explicou uso de um Map para comparar número e código.	Sim. Implementação ficou clara e coerente.
Votacaoservice.registarvoto	"Como impedir votos duplicados de um eleitor?"	Explicou que usou um Map com o id do eleitor e id do candidato	Sim

JUnit para Authenticationservice	"Como fazer teste JUnit para login válido e inválido?	Criou exemplo com @Test e Assertions.assertTrue / assertFalse.	Sim. Alguns teste JUnit 5 estavam corretos mas outros tivemos de corrigir manualmente.
----------------------------------	---	--	--

5. Reflexão Final

- Que vantagens e desvantagens identificaram no uso de LLM?

A vantagem é que acelera o desenvolvimento inicial, ajuda na decomposição de problemas e oferece alguns exemplos por onde começar. A desvantagem é que em sempre compreende o contexto completo. Às vezes foi necessário reformular o prompt para obter a resposta correta.

- Que tarefas foram simplificadas?

Estrutura de classes, criação de métodos repetitivos, testes JUnit, organização do projeto.

- Que tarefas continuam a requerer pensamento crítico?

Na resolução de problemas mais complexos ou menos óbvios.

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

Tornar-se-á cada vez mais responsável por orientar o LLM, validar código, garantir coerência entre componentes e manter a visão global do sistema.

5. Anexo Exemplo

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

#	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	Sérgio	Classe Administrador	S	N	R	Faz a função do requisito funcional01
2	Sérgio	Classe Administrador	S	N	R	Faz a função do requisito funcional02
3	Sérgio	Classe Eleitor	S	N	R	Faz a função do requisito funcional03
4	Rodrigo	Classe Votação / Administrador	S	N		Faz a função do requisito funcional04
5	Rodrigo	Classe Votação	S	N	R	Faz a função do requisito funcional05
6	Rodrigo	Classe main	S	N	R	Faz a função do requisito funcional06
6	Rodrigo	Classe Eleitor	S	N	R	Faz a função do requisito funcional07
7	Sérgio	Classe Eleitor	S	N	R	Garante que cada eleitor apenas pode votar uma vez
8	Rodrigo	Classe tests	S	N	R	Cria os testes unitarios para estas funcoes
9			...			