

Construção de um Modelo de Desenvolvimento/Geração de Código com IA

A presente dissertação tem como principal objetivo o desenvolvimento e avaliação de um modelo de geração automática de código suportado por técnicas de inteligência artificial, capaz de interpretar descrições de requisitos funcionais e convertê-las em código executável. Pretende-se explorar o potencial dos Large Language Models (LLMs) na automatização do ciclo de vida do software, desde a especificação de requisitos até à implementação de soluções programáticas. O sistema a desenvolver será concebido para compreender não apenas texto em linguagem natural, mas também representações visuais, como imagens, diagramas e esquemas.

A integração deste tipo de modelos visa aumentar a produtividade das equipas de desenvolvimento, reduzindo tarefas repetitivas e acelerando o processo de implementação de software. Contudo, reconhece-se que a adoção de LLMs neste contexto implica desafios técnicos e éticos, nomeadamente a qualidade e segurança do código gerado e a dependência de grandes volumes de dados de treino. Neste sentido, torna-se igualmente relevante compreender o impacto do prompt engineering, isto é, a formulação e estruturação das instruções dadas ao modelo, enquanto fator determinante na coerência e precisão das respostas produzidas. Deste modo, a investigação propõe-se não apenas à construção de um protótipo funcional, mas também à análise crítica das suas vantagens e limitações, contribuindo para uma compreensão mais sólida da aplicabilidade de LLMs multimodais na engenharia de software.

Para tal, este trabalho procurará aprofundar o estudo da robustez e fiabilidade do modelo, analisando de que forma diferentes tipos de input, textuais e visuais, bem como diferentes estratégias de prompt engineering, influenciam a consistência e qualidade do código produzido. O modelo desenvolvido gerará soluções recorrendo a Java/Spring para o backend e angular para o frontend, assegurando que o produto gerado respeita as boas práticas da empresa no que diz respeito à arquitetura, estrutura e segurança das aplicações. A sua eficácia, adaptabilidade e impacto prático serão avaliados em cenários reais de desenvolvimento, permitindo validar o contributo do sistema na melhoria da produtividade e qualidade do processo de programação.

CRONOGRAMA

Descrição das atividades	Out 25	Nov 25	Dez 25	Jan 26	Fev 26	Mar 26	Abr 26	Mai 26	Jun 26
Definição e revisão bibliográfica									
Análise de mercado e seleção de ferramentas									
Planeamento detalhado e estratégia de desenvolvimento									
Redação do documento preliminar									
Desenvolvimento do protótipo									
Avaliação e testes									
Análise e ajuste do modelo									
Redação do documento final									
Revisão final da tese e preparação da defesa									

Plano de Trabalhos

1. Definição e revisão bibliográfica

Nesta fase será realizada uma pesquisa aprofundada sobre o estado da arte na área de Large Language Models (LLMs), geração automática de código e engenharia de prompts. Serão analisados artigos científicos, relatórios

técnicos e frameworks relevantes, de modo a estabelecer uma base teórica sólida e identificar as principais abordagens existentes.

2. Análise de mercado e seleção de ferramentas

Será efetuado um levantamento das ferramentas e modelos disponíveis capazes de suportar geração de código assistida por IA. As opções identificadas serão comparadas com base na revisão bibliográfica realizada, destacando as suas principais vantagens e desvantagens. A seleção final dos modelos será orientada por critérios definidos em modelos de validação reconhecidos, garantindo a escolha das soluções mais adequadas para o desenvolvimento do protótipo.

3. Planeamento detalhado e estratégia de desenvolvimento

Com base na análise e comparação das ferramentas estudadas, será definida a estratégia de desenvolvimento a adotar, incluindo a arquitetura do sistema, a metodologia de implementação e os critérios de avaliação da solução proposta. A validação do sistema será orientada pelos modelos e métricas de avaliação identificados na fase anterior, garantindo uma análise consistente da qualidade e desempenho do protótipo.

4. Redação do documento preliminar

Será elaborado o documento preliminar da dissertação, contendo a contextualização do problema, objetivos do estudo, fundamentação teórica e metodologia proposta.

5. Desenvolvimento do Protótipo

Nesta fase será realizada a implementação do sistema proposto, contemplando o desenvolvimento da pipeline de entrada capaz de processar texto e imagens, a integração com o modelo LLM selecionado e a configuração do mecanismo de geração automática de código, garantindo a coerência sintática e semântica dos resultados obtidos.

6. Avaliação e testes

Nesta fase serão realizados testes comparativos com diferentes tipos de prompts, entradas textuais e visuais, de forma a avaliar a consistência, precisão e adaptabilidade do modelo.

7. Análise e ajuste do modelo

Com base nos resultados obtidos na fase de avaliação e testes (fase anterior), serão ajustados parâmetros do modelo e reformuladas estratégias de prompt engineering para melhorar a qualidade do código gerado. Esta fase visa otimizar a robustez e a fiabilidade do sistema.

8. Redação do documento final

Será desenvolvido o documento final da dissertação, integrando toda a análise teórica, metodologia, resultados experimentais e conclusões.

9. Redação Final da tese e preparação da defesa

Última fase dedicada à revisão formal do documento, normalização de referências, formatação final e preparação da apresentação e defesa do trabalho.