

Aluno(a): Hugo Rodrigues Torquato**Orientador(a): Heinrich Da Solidade Santos****Curso:** MBA em Engenharia de Software**Analizador de software C++ baseados em programação orientado a objetos****Introdução**

No contexto de programação orientada a objetos, as abstrações são fundamentais para organização e atribuição de responsabilidades. A compreensão da lógica é simples quando assuntos comumente conhecidos e/ou com poucos níveis de abstrações. Entretanto, o código se torna mais complexo e, por consequência, demanda mais esforço e tempo para compreendê-lo quando se trata de uma base de código extensa, desenvolvida durante vários anos, por vários profissionais e com foco em um conhecimento específico. Em uma ótica empresarial, este cenário implica em maiores investimentos para que novos engenheiros sejam incorporados à equipe por causa da complexidade, e adiciona um nível de abstração extra nas tomadas de decisão relacionados ao desenvolvimento do software por não ter uma visão geral do todo. Sob essa perspectiva, visualizar a organização de uma base de código clara e intuitivamente agrega valor, principalmente, na rotina das empresas desenvolvedoras de softwares.

A visualização de informações de um software auxilia o gerenciamento da base de código, o que, de forma secundária, reflete na maior manutenibilidade do produto. O livro Clean Code ilustra essa dinâmica ao comparar produtividade pelo tempo de existência do software: ao passo que a extensão e complexidade do código aumentam, especialmente sem planejamento, as chances de se transformar em um emaranhado de soluções improvisadas também aumentam. Nesse cenário, o autor argumenta que a produtividade para desenvolver novas funcionalidades e na manutenção do código declina drasticamente a médio e longo prazo, tornando-se quase impraticável e economicamente inviável.

O ponto de partida em projetos dessa natureza é a análise de código fonte, que é tema de estudos, a bastante tempo, para diversas áreas na computação: Otimização de compilador, como abordado por [1], que expõe benefícios ao realizar análises de hierarquia de classes como forma de resolver ambiguidades em chamadas de funções virtuais; Análises de qualidade de software, como abordado por [2], em alguns dos problemas frequentes na programação orientada a objetos; E até mesmo avaliar o quão eficiente os modelos de inteligência artificial baseados em linguagens (LLMs) estão quando comparado a programação funcional com orientada a objetos, abordado por [3], e que ressalta o quanto precisa ser melhorado quando avaliada especificidades ou em larga escala.

Das diversas aplicabilidades, soluções e produtos foram propostos com foco em prover uma melhor experiência durante e após o desenvolvimento de um software. As

ferramentas propostas podem ser introduzidas em diversos momentos do desenvolvimento de software, seja na etapa de codificação ou validação do que foi criado. [4] abordou em detalhe as benefícios gerais desse tipo de produto mas como forma de exemplificar temos: O SonarQube, uma plataforma de análise código com foco em qualidade e cobertura de testes; O *Clang Static Analyzer* uma ferramenta de análise nativa do LLVM aplicada ao C e C++; e *AddressSanitizer* como ferramenta para identificar erros de memória em tempo de execução.

Frente o exposto, ferramentas de visualizar e organização de uma base de código são bastante requisitadas em projetos de software. Por consequência, é tema de pesquisas na academia e também no desenvolvimento de novos produtos e soluções no âmbito privado. O conteúdo mais abordado no referencial teórico sobre o tema, ressalta uma investigação que é feita em nível de compiladores e nos ambientes de desenvolvimento integrado (IDEs), como forma de auxiliar o programador na escrita e otimização da compilação de código. Entretanto, o mapeamento estruturado e exposição organizada do código para discussões e tomada de decisão não vem sendo amplamente abordado e/ou com relevantes estudos e produtos para cobrir essa necessidade.

Objetivo

Este trabalho tem como objetivo desenvolver uma API capaz de realizar uma análise estática em códigos baseados em programação orientada a objetos, extrair informação sobre a organização das classes e exportar os resultados de forma estruturada.

Metodologia ou Material e Métodos

O presente trabalho tem como base a análise de código que, pela literatura, pode ser feito tanto por uma abordagem estática ou dinâmica. Um comparativo é apresentado por [4], que ressalt os aspectos positivos e os reveses de cada estratégia. O autor concui enfatizando o potencial positivo alcançado quando utilizas em conjunto. No contexto do projeto de pesquisa proposto, a análise estática revela-se como opção por depender somente do mais adequada o código fonte, sem a necessidade de compila-lo. O que permite uma expansão para demais linguagens de forma mais concisa e avalia a presente intenção do desenvolverdor por não considerar nenhuma otimização feita pelo compilador, o que é benéfico para compreensão da real arquitetura do software projetada para o sistema.

-> Link para falar dos parsers, que é uma forma legal para fazer essa análise estática

valiando o código fonte e não um intermediário já compilado.

Como abordado em Dilshan,

No tópico Metodologia ou Material e Métodos deve-se: detalhar como a pesquisa será conduzida; descrever os procedimentos que serão adotados para a coleta e análise de dados; e caracterizar a pesquisa em relação aos seus objetivos e à abordagem. Para os TCCs do MBA da USP/Esalq a pesquisa deve apresentar caráter aplicado e temática relacionada às áreas de desenvolvimento do curso, de modo que o(a) aluno(a) faça uso de ferramentas de metodologia de pesquisa e análise, bem como dos conhecimentos adquiridos ao longo do curso.

Entre os tipos de pesquisa recomendados para elaboração dos TCCs dos cursos do MBA USP/Esalq, têm-se: Pesquisa Experimental; Levantamento de Campo (“survey”); Estudo de Caso (único ou de múltiplos casos); Estudo de Caso-controle (ou pesquisa *ex-post-facto*); Pesquisa Participante; e, Pesquisa-Ação.

A metodologia escolhida determina quais são os dados (informações) necessários para a sua aplicação. Como técnicas e instrumentos para obtenção de informações e coleta de dados aceitos, menciona-se a Entrevista, o Questionário, a Observação, a Documental e o Levantamento de Dados Primários ou Secundários. As Pesquisas Bibliográficas, ou seja, baseadas exclusivamente em dados não experimentais levantados na literatura especializada em uma determinada área de conhecimento, são consideradas e aceitas somente se elaboradas utilizando métodos de análise de metadados ou grupo de dados (análise de redes sociais, análise de agrupamento, análise de correspondência, análise de fatores e análise de regressão múltipla). Cabe ressaltar que a depender do objeto de estudo, é necessário que a pesquisa seja submetida e aprovada pelo Comitê de Ética em Pesquisa [CEP], o que deve ser feito nas fases iniciais do estudo e antes da coleta de dados.

Não deve ser divulgado o nome da instituição ou empresa onde foi realizado o estudo ou onde foram coletados os dados, bem como nome do(s) proprietário(s) ou dos participantes de pesquisa. Assim, esta informação deve ser substituída pela descrição detalhada do local do estudo, incluindo informações como cidade e estado em que o objeto de estudo está localizado, atividade, porte, número de funcionários, entre outros dados.

O título desta seção deve ser definido com base na pesquisa a ser realizada. Quando se tratar de Pesquisa Bibliográfica, Pesquisa Documental ou Pesquisa Descritiva, deve-se utilizar “Metodologia” e quando de Pesquisa Experimental “Material e Métodos”.

Atenção: antes de enviar o arquivo para o Sistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.

Resultados Esperados

É planejado obter, com a conclusão da pesquisa, uma API capaz de realizar uma análise estática em códigos C++ e retornar um conjunto de dados estruturados em formato .json.

Nesse primeiro momento, a análise estática está focada na hierarquia entre as classes, de forma a estruturar seu relacionamento em um grafo capaz de agregar as informações e relações entre as classes.

Também será abordado uma discussão sobre a arquitetura escolhida para o desenvolvimento do software, de forma a explorar os benefícios dos padrões escolhidos frente as necessidades do presente projeto.

Cronograma de Atividades

O Cronograma de Atividades é o planejamento e organização da pesquisa e da escrita do TCC, o qual deve ser elaborado considerando as entregas das etapas do TCC estipuladas pela coordenação do curso.

Atenção: antes de enviar o arquivo para o Sistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.

Atividades planejadas	Mês									

Referências

Nas Referências deve ser elencado todas as obras utilizadas para a elaboração do Projeto de Pesquisa e normatizadas conforme o que se pede no “[Manual de Instruções e Normas para Trabalhos de Conclusão de Curso](#)”.

Atenção: antes de enviar o arquivo para o Sistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.