

# ADAPTAÇÃO E AVALIAÇÃO DE MERGE TEXTUAL BASE- ADO EM SEPARADORES SINTÁTICOS PARA A LINGUAGEM HASKELL

ADAPTATION AND EVALUATION OF TEXTUAL MERGE BASED ON  
SYNTACTIC SEPARATORS FOR HASKELL LANGUAGE

**Seu Nome Aqui**

seu.email@discente.ifpe.edu.br

**Nome do Orientador**

email.orientador@ifpe.edu.br

---

## RESUMO

O desenvolvimento colaborativo de software depende de ferramentas de controle de versão para integrar modificações paralelas. As ferramentas tradicionais de merge não estruturado (baseadas em linhas) frequentemente reportam falsos conflitos, impactando a produtividade. Abordagens baseadas em Árvore Sintáticas Abstratas (AST) reduzem esses conflitos, mas possuem alto custo de implementação por linguagem. Recentemente, foi proposto o CSDiff, uma abordagem textual baseada em separadores sintáticos, avaliada originalmente em Java. Este trabalho propõe a adaptação e avaliação desta técnica para Haskell, uma linguagem funcional com características sintáticas distintas, como a regra de layout e operadores customizados. O objetivo é verificar se a eficácia observada em linguagens imperativas se mantém no paradigma funcional.

Palavras-chave: Merge de Software. Haskell. Conflitos de Merge. Separadores Sintáticos.

---

## 1 INTRODUÇÃO

O desenvolvimento de software moderno é inerentemente colaborativo, exigindo que múltiplos desenvolvedores trabalhem simultaneamente nos mesmos artefatos de código. Para gerenciar essas modificações paralelas, Sistemas de Controle de Versão (VCS), como o Git, utilizam ferramentas de *merge* para integrar as alterações. A técnica predominante na indústria é o *merge* não estruturado (textual), que compara arquivos linha a linha (e.g., algoritmo *diff3*).

Embora amplamente utilizadas, as ferramentas baseadas em linhas apresentam limitações significativas. Elas frequentemente reportam conflitos falsos (falsos positivos) quando modificações ocorrem na mesma linha ou em linhas adjacentes, mesmo

que tais mudanças não interfiram semanticamente entre si (**cavalcanti2019impact**). Isso é particularmente crítico em linguagens que encorajam construções concisas ou onde a formatação sintática é densa.

Para mitigar esse problema, pesquisadores propuseram ferramentas de *merge* estruturado e semiestruturado, que utilizam a estrutura sintática (AST) do programa para realizar a integração (**apel2011semistructured**). No entanto, essas ferramentas exigem a construção de analisadores (*parsers*) complexos para cada linguagem, dificultando sua adoção generalizada.

Recentemente, **clementino2021textual** propuseram uma abordagem híbrida denominada *Custom Separators Diff* (CSDiff). Esta técnica mantém a simplicidade do processamento textual, mas utiliza separadores sintáticos específicos da linguagem (como ponto e vírgula ou chaves em Java) para segmentar o código antes da comparação, simulando o comportamento de ferramentas estruturadas com menor custo computacional.

## 1.1 Problema de Pesquisa

Enquanto o CSDiff demonstrou resultados promissores para a linguagem Java, sua eficácia em linguagens com paradigmas e sintaxes radicalmente diferentes, como **Haskell**, ainda não foi explorada. Haskell é uma linguagem puramente funcional que utiliza indentação significativa (*layout rule*) e faz uso extensivo de operadores customizados e assinaturas de tipos complexas em linhas únicas.

Essas características representam um desafio para o *merge* tradicional, pois pequenas alterações em definições de tipos ou em *pattern matching* podem ser interpretadas erroneamente como conflitos físicos. Portanto, este trabalho investiga se a abordagem baseada em separadores pode ser adaptada com sucesso para Haskell.

As seguintes Questões de Pesquisa (QPs), adaptadas do estudo original de **clementino2021textual**, norteiam este trabalho:

- **QP1:** A adaptação do CSDiff para Haskell reduz a quantidade total de conflitos reportados em comparação ao *merge* padrão (Diff3)?
- **QP2:** A ferramenta adaptada reduz a quantidade de cenários de *merge* com conflitos?
- **QP3:** A ferramenta adaptada reduz a quantidade de falsos conflitos (falsos positivos) em comparação ao Diff3?
- **QP4:** A ferramenta adaptada introduz integrações incorretas (falsos negativos) em comparação ao Diff3?

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Adaptar a técnica de *merge* textual baseado em separadores sintáticos para a linguagem Haskell e avaliar sua eficácia em comparação ao algoritmo tradicional *diff3*.

### 1.2.2 Objetivos Específicos

- Identificar os separadores sintáticos da linguagem Haskell mais relevantes para a resolução de conflitos (e.g., ::, ->, =>, =, |);
- Implementar um protótipo da ferramenta CSDiff adaptado para suportar a sintaxe e as peculiaridades do Haskell (como comentários e indentação);
- Coletar um conjunto de cenários de *merge* reais a partir de repositórios de software livre em Haskell;
- Executar um experimento comparativo entre o CSDiff adaptado e o *diff3*, analisando as métricas de conflitos, falsos positivos e falsos negativos.

## 2 FUNDAMENTAÇÃO TEÓRICA

## 3 METODOLOGIA

## 4 RESULTADOS PRELIMINARES (OU ESPERADOS)

## 5 CONCLUSÃO