

## A) CAPÍTULO — Validação Experimental do Core

Você pode copiar este texto direto para o relatório/TCC.

### 6. Validação Experimental do Core

A validação experimental do núcleo computacional (core) do projeto HyperVLM teve como objetivo verificar propriedades fundamentais de correção, robustez e previsibilidade da execução, sem dependência de interfaces gráficas, motores externos ou aplicações finais.

Todos os experimentos foram conduzidos utilizando exclusivamente o core isolado, compilado como biblioteca estática e executado por meio de uma ferramenta de linha de comando dedicada.

#### 6.1 Metodologia de Validação

A validação foi estruturada em três classes de testes:

Isolamento estrutural

Execução funcional mínima

Determinismo e tratamento de falhas

O ambiente de testes consistiu em sistema Linux, compilador GNU C++ e sistema de build CMake, com execução direta via terminal.

#### 6.2 Isolamento do Core

O core foi compilado de forma isolada, sem inclusão de módulos de renderização, path tracing ou motores gráficos. Para isso, foi criado um alvo específico de build contendo apenas os módulos essenciais:

Análise léxica (Lexer)

Análise sintática (Parser)

Construção de representação intermediária (IR)

Máquina virtual (VM)

Ambiente de execução (DRE)

A compilação isolada foi concluída com sucesso, demonstrando que o core não depende de componentes externos para sua construção ou execução.

Resultado: isolamento estrutural comprovado.

#### 6.3 Execução Funcional

Um programa mínimo válido foi utilizado como entrada para o core. O processo envolveu duas fases distintas:

Compilação da entrada para um artefato intermediário

Execução controlada pela máquina virtual

A execução apresentou comportamento incremental, com avanço explícito por etapas da VM, sem falhas ou comportamentos indefinidos.

Resultado: execução funcional mínima validada.

#### 6.4 Determinismo

O mesmo programa de entrada foi executado múltiplas vezes sob as mesmas condições. As saídas produzidas foram comparadas byte a byte.

Não foram observadas diferenças entre as execuções sucessivas.

Resultado: o core apresenta comportamento determinístico sob entradas equivalentes.

#### 6.5 Tratamento de Erros

Entradas sintaticamente inválidas foram fornecidas ao sistema com o objetivo de testar robustez. O core detectou o erro durante a fase de análise sintática e abortou a execução de forma segura, apresentando mensagem clara e controlada.

Não ocorreram falhas críticas, travamentos ou corrupção de estado.

Resultado: falhas são tratadas de forma previsível e segura.

#### 6.6 Invariantes Validadas

A partir dos experimentos, foram confirmadas as seguintes invariantes do core:

Determinismo da execução

Separação entre compilação e execução

Rejeição segura de entradas inválidas

Execução incremental e controlada

Independência de interfaces e motores externos

Essas propriedades qualificam o core como um núcleo computacional confiável, apto a servir como base para extensões futuras.