

Universidade Tiradentes
Ciência da Computação

Alan Reis Anjos
Anthony Ramos dos Santos
José Henrique Oliveira de Carvalho

Relatório da Implementação da Simulação e execução do código
Disciplina de Compiladores

Aracaju - SE
Novembro - 2025

Universidade Tiradentes

Ciência da Computação

Alan Reis Anjos
Anthony Ramos dos Santos
José Henrique Oliveira de Carvalho

Relatório da Implementação da Geração de Código Intermediário

Disciplina de Compiladores

Relatório sobre a **implementação da etapa de simulação do código gerado** apresentado como requisito parcial para a avaliação da disciplina de **Compiladores**, ministrada pela professora **Layse**, no curso de Ciência da Computação da **Universidade Tiradentes**.

Aracaju - SE
Novembro - 2025

Sumário

1	INTRODUÇÃO	5
2	JUSTIFICATIVA	6
3	OBJETIVOS	7
3.1	Objetivos Específicos	7
4	METODOLOGIA	8
4.1	Processo de Simulação	8
4.2	Ferramentas Utilizadas	8
4.3	Integração com o Compilador	9
5	RESULTADOS E DISCUSSÕES	10
5.1	Resultados Obtidos	10
5.2	Análise dos Comportamentos	10
5.3	Discussão	10
6	CONSIDERAÇÕES FINAIS	12
7	REFERÊNCIAS	13

1 INTRODUÇÃO

A linguagem de domínio específico **QuizLang** foi desenvolvida com o objetivo de permitir a definição estruturada de questionários, possibilitando que quizzes completos sejam descritos por meio de uma sintaxe clara e formal. As etapas anteriores do compilador — análise léxica, sintática, semântica e geração de código intermediário — asseguram que o conteúdo do quiz seja validado e traduzido para uma representação interna consistente.

A etapa apresentada neste relatório aborda a **simulação da execução do quiz**, que corresponde à fase final do fluxo de funcionamento do sistema. Ao contrário das fases anteriores, que lidam com interpretação e validação, a simulação é responsável por transformar o quiz definido no arquivo de entrada em uma experiência interativa com o usuário. É por meio da simulação que o quiz "ganha vida", permitindo que perguntas sejam exibidas, respostas sejam coletadas e o tempo limite seja respeitado conforme descrito pelo autor do quiz.

A simulação utiliza o código intermediário previamente gerado pelo compilador, garantindo que toda a lógica declarada na QuizLang seja preservada, incluindo metadados, seções, tipos de questões e suas regras específicas. Dessa forma, a simulação representa a aplicação prática da linguagem e demonstra, de forma concreta, sua utilidade e capacidade de modelar quizzes executáveis.

Este relatório descreve o funcionamento, a estrutura e o papel da simulação na arquitetura geral do sistema, destacando como ela interpreta o código intermediário e proporciona uma interação real com o usuário final.

2 JUSTIFICATIVA

A implementação da simulação da execução do quiz definido em **QuizLang** é uma etapa essencial para demonstrar, de forma prática, a aplicabilidade da linguagem desenvolvida. Embora as etapas de análise léxica, sintática, semântica e geração de código intermediário assegurem a correção estrutural e lógica do quiz, é somente por meio da simulação que o sistema comprova sua utilidade concreta, permitindo a interação direta entre o usuário e o conteúdo definido na DSL.

A simulação torna possível validar não apenas a integridade do código intermediário gerado pelo compilador, mas também seu comportamento em tempo de execução. Ela permite observar como perguntas são apresentadas, como respostas são coletadas e como as regras específicas de cada tipo de questão se manifestam em uma situação real de uso. Além disso, a simulação possibilita verificar se o fluxo definido pelo autor do quiz — incluindo seções, ordem das perguntas e restrições de tempo — está sendo respeitado corretamente.

Do ponto de vista pedagógico, esta etapa se justifica por demonstrar o ciclo completo de desenvolvimento de uma linguagem de domínio específico, desde sua definição até sua execução prática. Ela evidencia a importância da integração entre as fases do compilador e reforça o entendimento dos estudantes sobre a relação entre representação formal, tradução e aplicação final.

Em síntese, a simulação não apenas valida o funcionamento técnico do compilador, mas também comprova a efetividade da QuizLang como ferramenta para criação de questionários interativos, completando o propósito do projeto e ampliando seu potencial de utilização.

3 OBJETIVOS

Implementar e analisar a simulação da execução de quizzes definidos na linguagem **QuizLang**, utilizando o código intermediário gerado pelo compilador para transformar a especificação textual do quiz em uma experiência interativa e funcional para o usuário final.

3.1 Objetivos Específicos

- Interpretar o código intermediário produzido pelo compilador da QuizLang;
- Implementar a lógica de apresentação de perguntas e coleta de respostas;
- Controlar o tempo total do quiz, respeitando o limite definido na DSL;
- Aplicar as validações específicas de cada tipo de questão (múltipla escolha, numérica e discursiva);
- Registrar respostas, resultados e interações do usuário;
- Demonstrar, por meio da simulação, a aplicação prática completa da linguagem desenvolvida.

4 METODOLOGIA

A metodologia adotada para a simulação segue o fluxo lógico definido pela arquitetura do compilador da linguagem **QuizLang**. A simulação foi desenvolvida com base no código intermediário gerado na fase anterior, assegurando que toda a estrutura, regras e validações semânticas já estivessem consolidadas antes da execução.

4.1 Processo de Simulação

A implementação da simulação foi dividida em etapas:

1. **Leitura do código intermediário:** O simulador carrega a representação intermediária do quiz em formato estruturado (tipicamente JSON).
2. **Inicialização do ambiente de execução:** O sistema carrega metadados como título e tempo limite, e prepara o fluxo de seções e questões.
3. **Apresentação sequencial das questões:** Cada seção e suas questões são exibidas ao usuário na ordem definida na DSL.
4. **Coleta das respostas:** A interação varia conforme o tipo de questão:
 - **mcq:** seleção de opção;
 - **discursiva:** entrada textual aberta;
 - **numérica:** valor numérico com verificação de intervalo.
5. **Validação imediata:** Questões de múltipla escolha e numéricas são verificadas automaticamente.
6. **Registro dos dados:** Cada resposta é armazenada para análise posterior.
7. **Encerramento do quiz:** O sistema encerra ao final das questões ou quando o tempo limite é atingido.

4.2 Ferramentas Utilizadas

A simulação foi implementada em **Python**, mantendo consistência com o restante do compilador. A manipulação da estrutura intermediária foi feita por meio de dicionários e listas, oferecendo flexibilidade e compatibilidade com a exportação em JSON.

4.3 Integração com o Compilador

A simulação depende das fases anteriores do compilador. Somente quizzes validados sintática e semanticamente podem ser executados, assegurando que:

- não haja identificadores duplicados;
- intervalos numéricos sejam válidos;
- respostas de múltipla escolha estejam entre as opções disponíveis.

Essa integração garante que a simulação opere sobre dados corretos e bem estruturados.

5 RESULTADOS E DISCUSSÕES

A simulação demonstrou que o sistema baseado na linguagem **QuizLang** é capaz de transformar uma descrição textual formal em uma experiência prática interativa. Os testes realizados com arquivos como *teste_ok.txt* e *teste.txt* permitiram avaliar o comportamento da simulação em diferentes cenários.

5.1 Resultados Obtidos

Ao executar quizzes válidos, observou-se que:

- o título e o tempo limite foram corretamente exibidos ao usuário;
- todas as seções foram percorridas na ordem declarada;
- cada tipo de questão foi apresentado adequadamente;
- as validações automáticas funcionaram conforme esperado;
- o fluxo geral do quiz permaneceu fiel ao definido na DSL.

Para quizzes contendo inconsistências, a simulação não foi iniciada, confirmando a dependência adequada das fases anteriores do compilador.

5.2 Análise dos Comportamentos

A execução das questões de múltipla escolha demonstrou funcionamento completo, com identificação correta da resposta fornecida e comparação com a resposta declarada.

As questões numéricas apresentaram respostas corretamente verificadas contra os intervalos permitidos, reforçando a eficácia das validações semânticas.

As questões discursivas permitiram respostas abertas, mostrando que o simulador suporta interação textual sem restrições.

5.3 Discussão

A simulação exibiu comportamento estável e consistente. Os resultados confirmam que:

- a representação intermediária está adequada para uso prático;

- o compilador fornece todos os dados necessários para execução;
- a simulação cumpre o papel de validar o uso real da linguagem.

Além disso, a simulação demonstrou como a QuizLang pode ser usada além de um contexto acadêmico, podendo evoluir para uma ferramenta real de criação e execução de quizzes.

6 CONSIDERAÇÕES FINAIS

A simulação da execução de quizzes definidos na **QuizLang** representa a conclusão prática do ciclo de desenvolvimento do compilador. Por meio dela, é possível observar o funcionamento concreto da linguagem, validando tanto a estrutura definida pelo usuário quanto as análises realizadas pelo compilador.

A simulação demonstrou ser eficaz ao interpretar o código intermediário e convertê-lo em uma interação real, respeitando o tempo limite, a organização por seções e as regras específicas de cada tipo de questão. Esse processo confirma a solidez do projeto e a maturidade do compilador desenvolvido.

Além de cumprir sua função acadêmica, a simulação evidencia o potencial de expansão da QuizLang para ambientes reais, como plataformas de ensino e sistemas de avaliação automatizada.

Como trabalhos futuros, pode-se destacar:

- geração de relatórios automáticos ao final da simulação;
- criação de uma interface gráfica;
- integração com bancos de dados;
- possibilidade de geração automática de quizzes a partir de arquivos externos.

Assim, a etapa de simulação reforça o entendimento completo do processo de compilação e consolida o uso da QuizLang como uma DSL funcional, prática e extensível.

7 REFERÊNCIAS

AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores: princípios, técnicas e ferramentas**. 2. ed. São Paulo: Pearson, 2009.

PARR, Terence. **The Definitive ANTLR 4 Reference**. 2. ed. Dallas: Pragmatic Bookshelf, 2013. Disponível em: <https://www.antlr.org/>. Acesso em: 13 nov. 2025.

ANTLR. **ANTLR 4 Documentation**. Disponível em: <https://github.com/antlr/antlr4/blob/master/doc/index.md>. Acesso em: 13 nov. 2025.