



**Universidade Católica do
Salvador**

**VINICIUS DOS SANTOS CANDEIAS,
CRISTIANO DA SILVA OLIVEIRA FILHO,
ENZO SANTANA DE ALMEIDA,
RODRIGO LISBOA FIUZA E SILVA**

PROJETO CONCEITUAL

**Salvador
2022**

**VINICIUS DOS SANTOS CANDEIAS,
CRISTIANO DA SILVA OLIVEIRA FILHO,
ENZO SANTANA DE ALMEIDA
RODRIGO LISBOA FIUZA E SILVA**

PROJETO CONCEITUAL

Este projeto, apresentado como requisito parcial de avaliação da disciplina Compiladores, da Universidade Católica do Salvador.

Orientador: Osvaldo Requião Melo

**Salvador
2022**

Funcionamento do Static Checker:

Um verificador estático deve ser capaz de verificar qualquer texto fonte escrito na linguagem ORMPlus2022-2. O verificador estático será responsável por realizar a análise léxica e a análise parcial. Inicialmente, ele deve receber entradas de texto fonte escritas no formato ASCII com extensão .222. O programa executará então rotinas de análise léxica e sintática. Depois de concluir essas 3 etapas, o software gerará 2 tipos de arquivos, um é .LEX, que é o relatório de análise léxica, e o outro é .TAB, que contém o relatório da tabela de símbolos.

Funcionamento isolado do analisador léxico:

O analisador léxico pegará o texto fonte e percorrerá todos os caracteres do texto da seguinte forma, identificando os átomos e adicionando-os à lista de átomos encontrados. Você também deve ser responsável por verificar a lista de átomos encontrados e verificar se o átomo encontrado é um átomo da linguagem, em caso afirmativo, você deve verificar o tipo e o código do átomo, adicioná-lo à tabela e a linha em que está, em caso negativo serão salvos, juntamente com suas linhas na lista de caracteres e átomos inválidos.

Funcionamento isolado da tabela de símbolos:

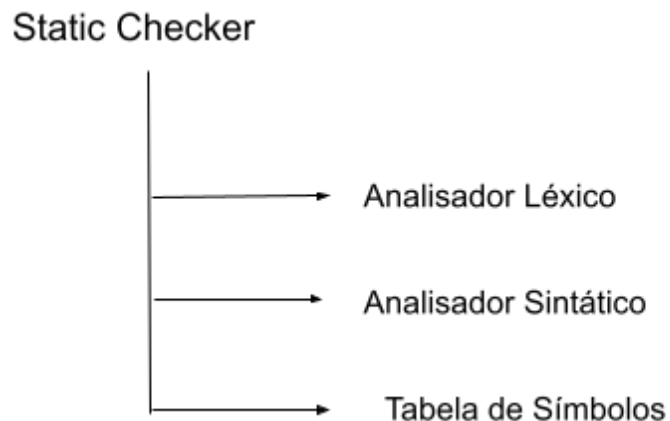
A tabela de símbolos deve conter, em uma tabela separada, uma lista apenas de átomos e palavras identificadoras e símbolos reservados para a linguagem ORMPlus-2022-2. O número de entrada deve ser salvo com o identificador do índice de armazenamento. O código atômico correspondente ao tipo de símbolo encontrado deve sempre ser preservado.

Funcionamento do analisador sintático:

O analisador deve ler todos os caracteres do texto fonte, ignorando letras maiúsculas e minúsculas, preservando átomos no texto fonte. O texto de origem pode consistir em qualquer sequência de caracteres, sejam eles válidos ou inválidos para o linguagem.

Estrutura do Projeto:

O projeto a ser desenvolvido terá três módulos. O primeiro, será o do analisador sintático, sendo responsável, também, pela entrada e saída do projeto. O segundo módulo será o do analisador léxico, responsável por receber e tratar o texto fonte. E por fim o módulo da tabela de símbolos.



Dificuldades do projeto:

Sobre as dificuldades enfrentadas no projeto, tivemos que refazer algumas vezes o autômato bruto e ótimo, por conta da ferramenta do excel, que não é tão apropriada e pela quantidade de estados presentes e símbolos reservados, no autômato bruto. Erramos a quantidade de estados j e tivemos que refazer algumas vezes o trabalho. A quantidade de tempo para entregar todas as etapas e o código final impactou no resultado esperado. Um membro da nossa equipe desistiu no meio do processo e acabou impactando nas etapas algumas entregas feitas em atraso e incompletas. Com esses fatos expostos, demos nosso máximo para fazer o melhor trabalho possível tendo em evidência nossas limitações, as quais foram citadas neste previamente neste texto.

Instruções para execução do StaticChecker

- Java 11 instalado para executar o projeto;
- No diretório do projeto adicione o seu arquivo para executar o arquivo final, só rodar a classe AnalisadorSintatico.java e obter o relatório .LEX e .TAB;
- No console, digite o nome do "seu arquivo" Exemplo: "test" que você quer segue imagem;

- Para executar o StaticChekerE02-2.7.3-jar no terminal digite (java -jar StaticChekerE02-2.7.3-jar)

```
jarvis@jarvis-b450mds3h:target$ cd ..
jarvis@jarvis-b450mds3h:E02LEXI$ ls
Entrega Final Projeto conceitual .pdf'  StaticChekerE02-2.7.3.jar
pom.xml  target
src  test.222
jarvis@jarvis-b450mds3h:E02LEXI$ java -jar StaticChekerE02-2.7.3.jar
Digite o nome do arquivo
test
Output Lexico
Tomos 4
jarvis@jarvis-b450mds3h:E02LEXI$
```

