



**Universidade Federal do Espírito Santo - Centro Tecnológico**  
**Departamento de Informática**  
**Compiladores (INF09281) – Turma 02**

**RELATÓRIO CP2**

**Gabriel Soares Xavier**  
**Mikaella Ferreira da Silva**

**Agosto/2021**

## 1 Objetivos

- Produção de um analisador semântico utilizando o parser e o lexer do CP1.
- Tratamento de tipos entre outros fatores que remetem ao analisador semântico.
- Gerar uma AST a partir do código passado para o analisador.

## 2 O que foi feito

Primeiramente, algumas simplificações foram realizadas, logo abaixo, vamos listar quais são as possíveis operações que são aceitas pelo compilador até o momento.

### 1) Tipos

- a) **int, double, String, List, void e bool.** Descartamos quaisquer formas de tipagem dinâmica, como o dart é misto, consideramos apenas as estáticas, a fins de simplificação.
  - i) Estamos considerando apenas listas de uma dimensão, também para simplificar o trabalho, ou seja podemos usar : List <int> x = [1,2,3], mas não podemos usar List x = [[1,2,3]] ou List <List <int> > x = [[1,2,3]].
  - ii) Listas são declaradas com a explicitude dos tipos, exemplo: List <type >
  - iii) Não existe possibilidade de criar listas heterogêneas e isso inclui adicionar inteiro a uma lista de doubles.

### 2) Declarações de Variáveis

- a) Global e Local
- b) Podem ser declaradas com valores ou sem
- c) Podem ser declaradas em cadeia, como : int x, y=5,z;
- d) Deve ser explicitamente declarada com o tipo

### 3) Declarações de funções

- a) Global e Local
- b) Somente funções declaradas no topo do código ou dentro de outras funções, sendo declaradas sempre antes do seu uso.
- c) Qualquer tipo de retorno
- d) Sem vários argumentos
- e) O comando "return;" só é permitido em funções de retorno void.

### 4) Loops

- a) For
- b) While
- c) Do while

### 5) Estrutura de escolha

- a) If-Then-Else

## **6) Atribuições**

- a) =
- b) \*=
- c) +=
- d) -=
- e) /=

## **7) Operadores Aritméticos**

- a) +
  - i) Para Listas e Strings, é concatenação.
- b) -
- c) /
- d) \*
- e) ~/ (retorno inteiro da divisão)
- f) %

## **8) Operadores de Comparação**

- a) >
- b) <
- c) >=
- d) <=
- e) ==
- f) !=

## **9) Operadores Lógicos**

- a) ||
- b) &&
- c) !

## **10) Operadores de Incremento**

- a) ++
- b) --

## **11) Conversões**

- a) Int para double, apenas

Tentamos tratar quaisquer tipos de inconsistências e erros de semântica, todos os casos testes apresentam sucesso assim como o esperado.

Descartamos as demais funcionalidades, tais como classes, mixins, entre outros, devido a sua complexidade, o que iria atrapalhar o rendimento do trabalho.

### 3 Casos de Teste

- **casosSucesso**

Os casos de teste desta pasta, são como o nome sugere, os casos que sempre dão certos (é o que se espera). Dividido em 12 arquivos .dart, onde cada um indica um tipo de caso particular.

- a) **caso\_1.dart**

- i) Caso de teste de if

- b) **caso\_2.dart**

- i) Caso de operadores lógicos

- c) **caso\_3.dart**

- i) Caso de operadores de comparação

- d) **caso\_4.dart**

- i) Caso de while

- e) **caso\_5.dart**

- i) Caso de Do while

- f) **caso\_6.dart**

- i) Caso de declarações de variáveis

- g) **caso\_7.dart**

- i) Caso de operadores de atribuição

- h) **caso\_8.dart**

- i) Caso de for

- i) **caso\_9.dart**

- i) Caso de declaração de função com operadores aritméticos

- j) **caso\_10.dart**

- i) Caso de função em escopos diferentes

- k) **caso\_11.dart**

- i) Caso de declaração de função local dentro de função

- l) **caso\_12.dart**

- i) Caso de função com lista e "return;"

- **casosErro**

Esses são os casos com erros semânticos, segue a mesma ordem que mencionado anteriormente, entretanto existem casos testes com erros, que possuem erro comentados, é necessário ir no arquivo e comentar e descomentar códigos para gerar os possíveis erros.

## **4 MAKEFILE**

O makefile tem as seguintes opções:

### **1) Run**

- a) Roda um caso específico, escreva no makefile o nome do arquivo.

### **2) Runall**

- a) Roda todos os casos de sucesso existentes.

### **3) Runallerro**

- a) Roda todos os casos de erro existentes.

Em ambos os casos, a AST é impressa dentro de um arquivo, na pasta resultados, com o nome do arquivo de caso de teste, exemplo caso\_1.dot, além disso um png da AST também é gerado, exemplo caso\_1.png. Certamente isso só serve para os casos de sucesso.