

Avance 01 – Proyecto Para Lenguajes De Programación

Pruebas Realizadas Por Paul Douglas Bustos Mite

Figura 1

Ejemplos de líneas de código en Dart para pruebas

```
m.test('''
  addone += 1;
  subone -= 1;
  EXPERIENCIA = 'EXP';
  List<String> lista = new List<>();
  Map<String, int> mapa = new Map<>();
  import 'package:utilities/utilities.dart';
  set right(double value) => left = value - width;
  static const initialCapacity = 16;
  typedef IntList = List<int>;
''')
```

Figura 2

Primera parte de los resultados obtenidos en las pruebas

```
LexToken(ID, 'addone', 2, 5)
LexToken(PLUS, '+', 2, 12)
LexToken(EQUAL, '=', 2, 13)
LexToken(DIGIT, '1', 2, 15)
LexToken(SEMICOLON, ';', 2, 16)
LexToken(ID, 'subone', 3, 22)
LexToken(MINUS, '-', 3, 29)
LexToken(EQUAL, '=', 3, 30)
LexToken(DIGIT, '1', 3, 32)
LexToken(SEMICOLON, ';', 3, 33)
LexToken(ID, 'EXPERIENCIA', 4, 39)
LexToken(EQUAL, '=', 4, 51)
LexToken(APOSTROPHE, '"', 4, 53)
LexToken(ID, 'EXP', 4, 54)
LexToken(APOSTROPHE, '"', 4, 57)
LexToken(SEMICOLON, ';', 4, 58)
```

Figura 3

Segunda parte de los resultados obtenidos en las pruebas

```
LexToken(DIAMOND_OPERATOR, 'List<String>', 5, 64)
LexToken(ID, 'lista', 5, 77)
LexToken(EQUAL, '=', 5, 83)
LexToken(ID, 'new', 5, 85)
LexToken(ID, 'List', 5, 89)
LexToken(LESS_THAN, '<', 5, 93)
LexToken(GREATER_THAN, '>', 5, 94)
LexToken(LPARENTHESIS, '(', 5, 95)
LexToken(RPARENTHESIS, ')', 5, 96)
LexToken(SEMICOLON, ';', 5, 97)
LexToken(ID, 'Map', 6, 103)
LexToken(LESS_THAN, '<', 6, 106)
LexToken(ID, 'String', 6, 107)
LexToken(COMA, ',', 6, 113)
LexToken(ID, 'int', 6, 115)
LexToken(GREATER_THAN, '>', 6, 118)
```

Figura 4

Tercera parte de los resultados obtenidos en las pruebas

```
LexToken(ID, 'mapa', 6, 120)
LexToken(EQUAL, '=', 6, 125)
LexToken(NEW, 'new', 6, 127)
LexToken(DIAMOND_OPERATOR, 'Map<>()', 6, 131)
LexToken(SEMICOLON, ';', 6, 138)
LexToken(IMPORT, 'import', 7, 144)
LexToken(APOSTROPHE, '"', 7, 151)
LexToken(ID, 'package', 7, 152)
LexToken(COLON, ':', 7, 159)
LexToken(ID, 'utilities', 7, 160)
LexToken(SLASH, '/', 7, 169)
LexToken(ID, 'utilities', 7, 170)
```

Figura 5

Cuarta parte de los resultados obtenidos en las pruebas

```
LexToken(DOT, '.', 7, 179)
LexToken(ID, 'dart', 7, 180)
LexToken(APOSTROPHE, "'", 7, 184)
LexToken(SEMICOLON, ';', 7, 185)
LexToken(ID, 'set', 8, 191)
LexToken(ID, 'right', 8, 195)
LexToken(LPARENTHESIS, '(', 8, 200)
LexToken(ID, 'double', 8, 201)
LexToken(ID, 'value', 8, 208)
LexToken(RPARENTHESIS, ')', 8, 213)
LexToken(EQUAL, '=', 8, 215)
LexToken(GREATER_THAN, '>', 8, 216)
LexToken(ID, 'left', 8, 218)
LexToken(EQUAL, '=', 8, 223)
LexToken(ID, 'value', 8, 225)
LexToken(MINUS, '-', 8, 231)
```

Figura 6

Quinta parte de los resultados obtenidos en las pruebas

```
LexToken(ID, 'width', 8, 233)
LexToken(SEMICOLON, ';', 8, 238)
LexToken(STATIC, 'static', 9, 244)
LexToken(CONST, 'const', 9, 251)
LexToken(ID, 'initialCapacity', 9, 257)
LexToken(EQUAL, '=', 9, 273)
LexToken(DIGIT, '1', 9, 275)
LexToken(DIGIT, '6', 9, 276)
LexToken(SEMICOLON, ';', 9, 277)
LexToken(TYPEDEF, 'typedef', 10, 283)
LexToken(ID, 'IntList', 10, 291)
LexToken(EQUAL, '=', 10, 299)
LexToken(DIAMOND_OPERATOR, 'List<int>', 10, 301)
LexToken(SEMICOLON, ';', 10, 310)
```

Pruebas Realizadas Por Jorge Luis Valenzuela Bustos

Figura 7

Resultados obtenidos en las pruebas

```
>>null
LexToken(NULL, 'null',11,0)
>>return
LexToken(RETURN, 'return',11,0)
>>this
LexToken(THIS, 'this',11,0)
>>>true
LexToken(TRUE, 'true',11,0)
>>var
LexToken(VAR, 'var',11,0)
>>while
LexToken(WHILE, 'while',11,0)
>>List
LexToken(LISTA, 'List',11,0)
>>Set
LexToken(CONJUNTO, 'Set',11,0)
>>Map
LexToken(MAPA, 'Map',11,0)
>>==
LexToken(DOUBLEIGUAL, '==',11,0)
>><
LexToken(MENOR, '<',11,0)
>><=
LexToken(MENORIGUAL, '<=',11,0)
>>"Hola Mundo"
LexToken(CADENA, '"Hola Mundo"',11,0)
```

Pruebas Realizadas Por Douglas Javier Sabando Macías

Figura 8

Resultados obtenidos en las pruebas

```
continue
LexToken(CONTINUE, 'continue', 1, 0)
else
LexToken(ELSE, 'else', 1, 0)
break
LexToken(BREAK, 'break', 1, 0)
class
LexToken(CLASS, 'class', 1, 0)
final
LexToken(FINAL, 'final', 1, 0)
for
LexToken(FOR, 'for', 1, 0)
if
LexToken(IF, 'if', 1, 0)
new
LexToken(NEW, 'new', 1, 0)
print
LexToken(PRINT, 'print', 1, 0)
readLineSync
LexToken(READLINESYNC, 'readLineSync', 1, 0)
export
LexToken(EXPORT, 'export', 1, 0)
function
LexToken(FUNCTION, 'function', 1, 0)
get
LexToken(GET, 'get', 1, 0)
```