

**TECNOLÓGICO DE COSTA RICA**  
**Escuela de Ingeniería en Computación**  
**Proyecto de Ingeniería de Software**

**Profesora:**

María Estrada Sánchez

**Entrega 3:**

Analizador Contextual:

Plan de pruebas, validación y verificación.

**Estudiantes:**

Christian León Guevara - 2013371982

Gabriel Ramírez Ramírez - 201020244

**Fecha de entrega:**

20-01-2019

**Período Verano**  
**Cartago**

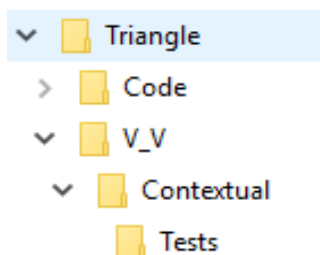
## Tabla de contenido

Introducción .....	3
1. Ejecución de pruebas de los casos propuestos. ....	4
2. Ejecución de las revisiones validación y verificación del código. ....	26

## Introducción

En este documento se presenta la ejecución de las pruebas de validación y verificación del Analizador Contextual para poder verificar cada uno de los casos que aquí se muestra es necesario confrontar el documento *Catálogo de Pruebas.docx* donde se establecen el objetivo y resultado que cada caso debe tener.

La ubicación de los archivos de los casos de prueba se encuentra en la carpeta Triangle/V\_V/Contextual/tests.



Listado de los casos de prueba:

arrAccContextualErrorModoGet.tri	procErrConstParamVar01.tri
arrAccContextualErrorModoPutint.tri	procErrConstParamVar02.tri
arrAccContextualErrorNParamsGet.tri	procErrConstParamVar03.tri
arrAccContextualErrorTipoPutint.tri	procErrDesconocido.tri
arrAccContextualOKEjecMal.tri	procErrFuncParam01.tri
constErr01.tri	procErrFuncParam02.tri
constErrDeclarationRepeat01.tri	procErrNumParams01.tri
constErrMayusYMinus01.tri	procErrNumParams02.tri
dirAccDeclaration01.tri	procErrParamInvisible.tri
dirAccErr01.tri	procErrParamsRepet01.tri
expressionErr01.tri	procErrProcParam01.tri
funcDeclaration01.tri	procErrProcParam02.tri
funcDeclaration02.tri	procErrTipoParam.tri
IfErrExpression01.tri	procErrTipoParam01.tri
IfErrExpression02.tri	procOKFuncParam01.tri
IfErrExpression03.tri	procOKParam.tri
IfErrExpression04.tri	varAssign.tri
IfErrExpression05.tri	varErrAssign.tri
noDeclarationErr01.tri	whileErrExpression01.tri
noDeclarationErr02.tri	whileErrExpression02.tri

## 1. Ejecución de pruebas de los casos propuestos.

### Caso 1. arrAccContextualErrorModoGet.tri

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/arrAccContextualErrorModoGet.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: const actual parameter not expected here 12..12
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle>
Triangle ./V_V/Contextual/Tests/arrAccContextualErrorModoGet.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Const actual parameter not expected here [12:12]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error haciendo referencia a la posición en el documento

Resultado de prueba

- La prueba es exitosa

### Caso 2. arrAccContextualModoPutInt

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/arrAccContextualErrorModoPutInt.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: var actual parameter not expected here 13..13
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle>
Triangle ./V_V/Contextual/Tests/arrAccContextualErrorModoPutInt.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Var actual parameter not expected here [13:20]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

### Caso 3. arrAccContextualNParamsGet

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/arrAccContextualErrorNParamsGet.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: too many actual parameters 12..12
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/arrAccContextualErrorNParamsGet.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Too many actual parameters [12:26]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

### Caso 4. arrAccContextualTipoPutInt

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/arrAccContextualErrorTipoPutInt.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: wrong type for var actual parameter 11..11
ERROR: wrong type for const actual parameter 13..13
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/arrAccContextualErrorTipoPutInt.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Wrong type for var actual parameter [11:16]
ERROR: Wrong type for const actual parameter [13:16]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 5. arrAccContextualOKEjecMal

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/arrAccContextualOKEjecMal.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/arrAccContextualOKEjecMal.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se no se presentan errores.

Resultado de prueba

- La prueba es exitosa

## Caso 6. constErr01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/constErr01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: LHS of assignment is not a variable 7..7
ERROR: "x" is not declared 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/constErr01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: LHS of assignment is not a variable [7:9]
ERROR: x is not a const or var Identifier [8:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan los mismos errores, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 7. constErrDeclarationRepeat01

Resultado en Java

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/constErrDeclarationRepeat01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: identifier "x" already declared 5..5
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleTriCompiler.exe ./V_V/Contextual/Tests/constErrDeclarationRepeat01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Identifier x already declared [6:2]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, tanto la columna como la fila están incorrectas, pero esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 8. constErrMayusYMinus01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/constErrMayusYMinus01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "X" is not declared 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleTriCompiler.exe ./V_V/Contextual/Tests/constErrMayusYMinus01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: X is not a const or var Identifier [7:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 9. dirAccDeclaration01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/dirAccDeclaration01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle
riangle ./V_V/Contextual/Tests/dirAccDeclaration01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se no se presentan errores.

Resultado de prueba

- La prueba es exitosa

## Caso 10. dirAccErr01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/dirAccErr01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: wrong type for const actual parameter 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/dirAccErr01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Wrong type for const actual parameter [7:13]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa



## Caso 11. expressionErr01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/expressionErr01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: wrong argument type for "+" 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle>Triangle ./V_V/Contextual/Tests/expressionErr01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Wrong argument type for + [8:15]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 12. funcDeclaration01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/funcDeclaration01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: duplicated formal parameter "par1" 4..4
ERROR: duplicated formal parameter "par2" 5..5
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle>Triangle ./V_V/Contextual/Tests/funcDeclaration01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Duplicated formal parameter par1 [4:45]
ERROR: Duplicated formal parameter par2 [5:45]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra columna esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

### Caso 13. funcDeclaration02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/funcDeclaration02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle>
Triangle ./V_V/Contextual/Tests/funcDeclaration02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se no se presentan errores.
- Resultado de prueba
- La prueba es exitosa

### Caso 14. IfErrExpression01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/IfErrExpression01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Boolean expression expected here 8..0
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle>
Triangle ./V_V/Contextual/Tests/IfErrExpression01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Boolean expression expected here [16:0]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.
- Resultado de prueba
- La prueba es exitosa

## Caso 15. IfErrExpression02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/IfErrExpression02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/IfErrExpression02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se no se presentan errores.

Resultado de prueba

- La prueba es exitosa

## Caso 16. IfErrExpression03

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/IfErrExpression03.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/IfErrExpression03.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se no se presentan errores.

Resultado de prueba

- La prueba es exitosa

## Caso 17. IfErrExpression04

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/IfErrExpression04.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Boolean expression expected here 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/IfErrExpression04.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Boolean expression expected here [11:0]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, tanto la columna como la fila están incorrectas, pero esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso18. IfErrExpression05

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/IfErrExpression05.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: incompatible limbs in if-expression 7..7
ERROR: Boolean expression expected here 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/IfErrExpression05.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Incompatible limbs in if-expression [7:31]
ERROR: Boolean expression expected here [11:0]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 19. noDeclaractionErr01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/noDeclaractionErr01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "a" is not declared 9..9
ERROR: LHS of assignment is not a variable 9..9
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/noDeclaractionErr01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: a is not a const or var Identifier [9:7]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 20. noDeclaractionErr02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/noDeclaractionErr02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "a" is not declared 10..10
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/noDeclaractionErr02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: a is not a const or var Identifier [10:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 21. procErrConstParamVar01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrConstParamVar01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: actual parameter is not a variable 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleTriCompiler .\V_V\Contextual\Tests/procErrConstParamVar01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Actual parameter is not a variable [8:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 22. procErrConstParamVar02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrConstParamVar02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: actual parameter is not a variable 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleTriCompiler .\V_V\Contextual\Tests/procErrConstParamVar02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Actual parameter is not a variable [8:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 23. procErrConstParamVar03

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrConstParamVar03.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: const actual parameter not expected here 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrConstParamVar02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Actual parameter is not a variable [8:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 24. procErrDesconocido

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrDesconocido.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "P" is not declared 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrDesconocido.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: P is not declared [7:3]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa



## Caso 25. procErrFuncParam01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrFuncParam01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "F" is not a const or var identifier 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrFuncParam01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: F is not a const or var Identifier [8:7]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 26. procErrFuncParam02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrFuncParam02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: const actual parameter not expected here 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrFuncParam02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Const actual parameter not expected here [8:9]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa



## Caso 27. procErrNumParams01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrNumParams01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: too many actual parameters 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrNumParams01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Too many actual parameters [7:12]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 28. procErrNumParams02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrNumParams02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: too few actual parameters 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrNumParams02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Too few actual parameters [7:6]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 29. procErrParamInvisible

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrParamInvisible.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "x" is not declared 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleTriCompiler .\V_V\Contextual\Tests/procErrParamInvisible.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: x is not a const or var Identifier [8:7]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 30. procErrParamsRepet01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrParamsRepet01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: duplicated formal parameter "x" 5..5
ERROR: wrong argument type for "*" 6..6
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleTriCompiler .\V_V\Contextual\Tests/procErrParamsRepet01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Duplicated formal parameter x [5:50]
ERROR: Wrong argument type for * [6:18]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 31. procErrProcParam01

Resultado en Java.

Al realizar la ejecución de este caso en el compilador de Java se presentó un error y no se realiza el análisis contextual por tal motivo no se ha colocado dicha evidencia.

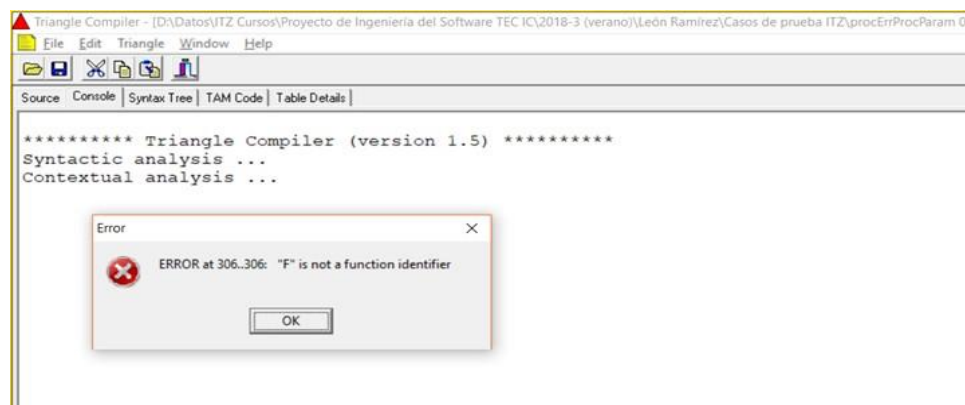
```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrProcParam01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "F" is not a function identifier 8..8
Exception in thread "main" java.lang.NullPointerException
    at Triangle.ContextualAnalyzer.Checker.visitConstActualParameter(Checker.java:469)
    at Triangle.AbstractSyntaxTrees.ConstActualParameter.visit(ConstActualParameter.java:27)
    at Triangle.ContextualAnalyzer.Checker.visitSingleActualParameterSequence(Checker.java:573)
    at Triangle.AbstractSyntaxTrees.SingleActualParameterSequence.visit(SingleActualParameterSequence.java:28)
    at Triangle.ContextualAnalyzer.Checker.visitCallCommand(Checker.java:113)
    at Triangle.AbstractSyntaxTrees.CallCommand.visit(CallCommand.java:29)
    at Triangle.ContextualAnalyzer.Checker.visitLetCommand(Checker.java:138)
    at Triangle.AbstractSyntaxTrees.LetCommand.visit(LetCommand.java:28)
    at Triangle.ContextualAnalyzer.Checker.visitProgram(Checker.java:743)
    at Triangle.AbstractSyntaxTrees.Program.visit(Program.java:27)
    at Triangle.ContextualAnalyzer.Checker.check(Checker.java:757)
    at Triangle.Compiler.compileProgram(Compiler.java:94)
    at Triangle.Compiler.main(Compiler.java:129)
```

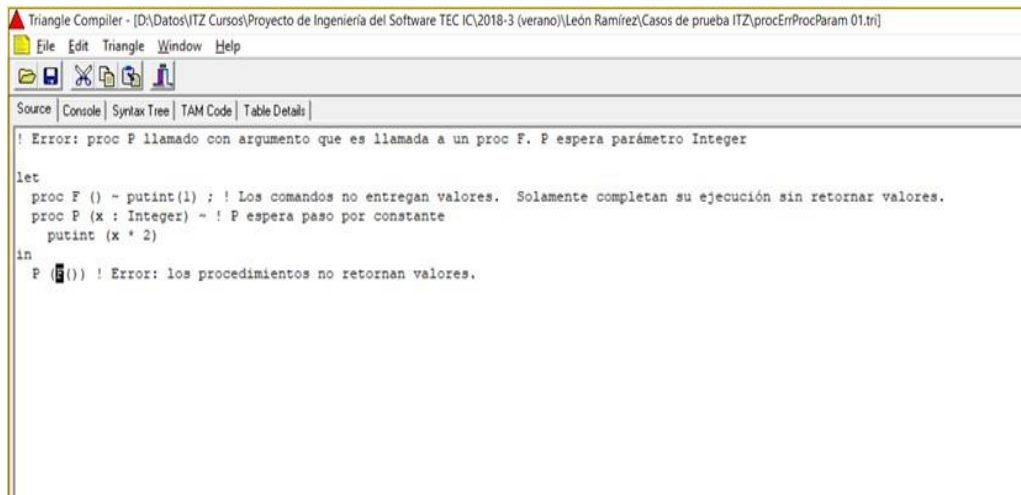
Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrProcParam01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: F is not a function Identifier [8:9]
Compilation was unsuccessful
```

Análisis de resultado

- En el compilador de Java se produce un error de memoria con este ejemplo por lo que no se pudo realizar la prueba. Este error también fue reproducido en el compilador de Pascal que existe.





The screenshot shows the Triangle Compiler IDE. The title bar indicates the file path: [D:\Datos\ITZ Cursos\Proyecto de Ingeniería del Software TEC IC\2018-3 (verano)\León Ramírez\Casos de prueba ITZ\procEnProcParam 01.tri]. The menu bar includes File, Edit, Triangle, Window, and Help. Below the menu bar is a toolbar with icons for file operations and compilation. The main window has tabs for Source, Console, Syntax Tree, TAM Code, and Table Details. The Source tab is active, displaying the following OCaml code:

```
! Error: proc P llamado con argumento que es llamada a un proc F. P espera parámetro Integer

let
  proc F () ~ putint(1) ; ! Los comandos no entregan valores. Solamente completan su ejecución sin retornar valores.
  proc P (x : Integer) ~ ! P espera paso por constante
    putint (x * 2)
in
  P (F()) ! Error: los procedimientos no retornan valores.
```

The error messages are in Spanish. The first error, at the top, states: "Error: proc P llamado con argumento que es llamada a un proc F. P espera parámetro Integer" (Error: proc P called with argument that is a call to a proc F. P expects parameter Integer). The second error, at the bottom, states: "Error: los procedimientos no retornan valores." (Error: the procedures do not return values).

- Prueba inválida
- Cabe rescatar que el compilador en Ocaml si realiza correctamente el analizador contextual.

## Caso 32. procErrProcParam02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrProcParam02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "F" is not a const or var identifier 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrProcParam02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: F is not a const or var Identifier [8:7]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 33. procErrTipoParam

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrTipoParam.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: wrong type for const actual parameter 7..7
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrTipoParam.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Wrong type for const actual parameter [7:9]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 34. procErrTipoParam01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procErrTipoParam01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "Entero" is not a const or var identifier 8..8
ERROR: const actual parameter not expected here 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procErrTipoParam.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Wrong type for const actual parameter [7:9]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 35. procOKFuncParam01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procOKFuncParam01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procOKFuncParam01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se no se presentan errores.

Resultado de prueba

- La prueba es exitosa

## Caso 36. procOKParam

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/procOKParam.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/procOKParam.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se presentan el mismo error, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## Caso 37. varAssign

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/varAssign.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
Code Generation ...
Compilation was successful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleT
riangle ./V_V/Contextual/Tests/varAssign.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
Compilation was successful
```

Análisis de resultado

- En ambos compiladores se no se presentan errores.

Resultado de prueba

- La prueba es exitosa



## Caso 38. varErrAssign

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/varErrAssign.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: assignment incompatibilty 10..10
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle\
Triangle ./V_V/Contextual/Tests/varErrAssign.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: Assignment incompatibility [10:11]
Compilation was unsuccessful
```

## Caso 39. whileErrExpression01

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/whileErrExpression01.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "n" is not declared 7..7
ERROR: Boolean expression expected here 7..0
ERROR: "n" is not declared 8..8
ERROR: "n" is not declared 8..8
ERROR: LHS of assignment is not a variable 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\Triangle\
Triangle ./V_V/Contextual/Tests/whileErrExpression01.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: n is not a const or var Identifier [7:12]
ERROR: Boolean expression expected here [9:0]
ERROR: n is not a const or var Identifier [8:7]
ERROR: n is not a const or var Identifier [8:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan los mismos errores, pero aun que hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador sintáctico no es responsable de ello.

Resultado de prueba

- La prueba es exitosa



## Caso 40. whileErrExpression02

Resultado en Java.

```
C:\0 Proyecto\Analizador-Contextual\Codigo\JavaTriCompiler>java Triangle.Compiler tests/whileErrExpression02.tri
***** Triangle Compiler (Java Version 2.1) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: "n" is not declared 7..7
ERROR: Boolean expression expected here 7..0
ERROR: "n" is not declared 8..8
ERROR: "n" is not declared 8..8
ERROR: LHS of assignment is not a variable 8..8
Compilation was unsuccessful.
```

Resultado en OCAML.

```
C:\Users\Christian\Documents\Ana_Contextual\Analizador-Contextual\Codigo\Código integrado con Sintactico\TriangleCompiler.exe ./V_V/Contextual/Tests/whileErrExpression02.tri
***** Triangle Compiler (Caml Version 1.0) *****
Syntactic Analysis ...
Contextual Analysis ...
ERROR: n is not a const or var Identifier [7:12]
ERROR: Boolean expression expected here [9:0]
ERROR: n is not a const or var Identifier [8:7]
ERROR: n is not a const or var Identifier [8:11]
Compilation was unsuccessful
```

Análisis de resultado

- En ambos compiladores se presentan los mismos errores, pero aunque hace referencia a otra fila esto se debe a cuando se leen los tokens del árbol. El analizador contextual no es responsable de ello.

Resultado de prueba

- La prueba es exitosa

## 2. Ejecución de las revisiones validación y verificación del código.

### 1.1 Lista de cotejo de legibilidad.

#### Checkel.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Presenta un encabezado en los archivos con los derechos de propiedad y los autores.	X		
Las importaciones se colocan al comienzo del archivo.	X		
El nombre de las funciones es significativo y descriptivo.	X		
El nombre de las variables e identificadores es significativo y descriptivo.	X		
Se mantiene el nivel de indentación de 2 espacios establecido en el estándar de estilos para Ocaml.	X		
Se respetó el ancho de 80 columnas dentro del código.	X		Algunas líneas se dejaron excediendo el ancho de 80 columnas.
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.	X		

## IdentificationTable.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Presenta un encabezado en los archivos con los derechos de propiedad y los autores.	X		
Las importaciones se colocan al comienzo del archivo.	X		
El nombre de las funciones es significativo y descriptivo.	X		
El nombre de las variables e identificadores es significativo y descriptivo.	X		
Se mantiene el nivel de indentación de 2 espacios establecido en el estándar de estilos para Ocaml.	X		
Se respetó el ancho de 80 columnas dentro del código.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.	X		

## IdEntry.mli

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Presenta un encabezado en los archivos con los derechos de propiedad y los autores.	X		Se describe los autores y fecha de modificación.
Las importaciones se colocan al comienzo del archivo.	X		
El nombre de las funciones es significativo y descriptivo.		X	
El nombre de las variables e identificadores es significativo y descriptivo.	X		Hace referencia al nombre de la entrada de la tabla de identificación.
Se mantiene el nivel de indentación de 2 espacios establecido en el estándar de estilos para Ocaml.		X	
Se respetó el ancho de 80 columnas dentro del código.		X	Es una única declaración por lo que se dejó en un ancho superior a las 80 columnas.
Se rompieron líneas de código muy extensas.		X	
Se utilizó la agrupación e indentación utilizando paréntesis.		X	

## Ast.mli

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Presenta un encabezado en los archivos con los derechos de propiedad y los autores.	X		
Las importaciones se colocan al comienzo del archivo.	X		
El nombre de las funciones es significativo y descriptivo.	X		
El nombre de las variables e identificadores es significativo y descriptivo.	X		
Se mantiene el nivel de indentación de 2 espacios establecido en el estándar de estilos para Ocaml.	X		En algunos casos no se respeta la indentación de 2 espacios establecida en el estándar.
Se respetó el ancho de 80 columnas dentro del código.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.		X	

## ErrorReporter.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Presenta un encabezado en los archivos con los derechos de propiedad y los autores.	X		
Las importaciones se colocan al comienzo del archivo.	X		
El nombre de las funciones es significativo y descriptivo.	X		
El nombre de las variables e identificadores es significativo y descriptivo.	X		Existe un error en el nombre de la función <code>write_x_m_l_errors</code> .
Se mantiene el nivel de indentación de 2 espacios establecido en el estándar de estilos para Ocaml.	X		
Se respetó el ancho de 80 columnas dentro del código.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.	X		

## IdentificationTablePrinter\_XML.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Presenta un encabezado en los archivos con los derechos de propiedad y los autores.	X		
Las importaciones se colocan al comienzo del archivo.	X		
El nombre de las funciones es significativo y descriptivo.	X		
El nombre de las variables e identificadores es significativo y descriptivo.	X		
Se mantiene el nivel de indentación de 2 espacios establecido en el estándar de estilos para Ocaml.	X		
Se respetó el ancho de 80 columnas dentro del código.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.		X	

## TreeDrawer.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Presenta un encabezado en los archivos con los derechos de propiedad y los autores.	X		
Las importaciones se colocan al comienzo del archivo.	X		
El nombre de las funciones es significativo y descriptivo.	X		
El nombre de las variables e identificadores es significativo y descriptivo.	X		Existe un error en el nombre de la función <code>write_x_m_l_tree</code> .
Se mantiene el nivel de indentación de 2 espacios establecido en el estándar de estilos para Ocaml.	X		
Se respetó el ancho de 80 columnas dentro del código.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.		X	



## 1.2 Lista de cotejo de comprensibilidad.

### Checkel.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Dispone un encabezado para las funciones y lo que estas realizan.	X		
El nombre de las funciones se establece mediante el estilo snake_case.	X		
El nombre de las variables e identificadores se establecen mediante el estilo snake_case.	X		
El nombre de los Constructores se establece mediante el estilo Snake_case.	X		
Los acrónimos técnicos se escribieron todo en mayúscula.		X	
Existe una estandarización de los nombres.	X		
Se mantuvo la indentación de 2 espacios.	X		
Se rompieron líneas de código muy extensas.	X		Algunas líneas de código superan el ancho de 80 columnas.
Se utilizó la agrupación e indentación utilizando paréntesis.	X		
Existen comentarios innecesarios.		X	
Existen comentarios de código que ha sido eliminado.		X	
Relación lógica del código con relación a la documentación.	X		

## IdentificationTable.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Dispone un encabezado para las funciones y lo que estas realizan.	X		
El nombre de las funciones se establece mediante el estilo snake_case.	X		
El nombre de las variables e identificadores se establecen mediante el estilo snake_case.	X		
El nombre de los Constructores se establece mediante el estilo Snake_case.	X		
Los acrónimos técnicos se escribieron todo en mayúscula.		X	
Existe una estandarización de los nombres.	X		
Se mantuvo la indentación de 2 espacios.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.	X		
Existen comentarios innecesarios.		X	
Existen comentarios de código que ha sido eliminado.		X	
Relación lógica del código con relación a la documentación.	X		

## IdEntry.mli

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Dispone un encabezado para las funciones y lo que estas realizan.		X	
El nombre de las funciones se establece mediante el estilo snake_case.		X	
El nombre de las variables e identificadores se establecen mediante el estilo snake_case.	X		
El nombre de los Constructores se establece mediante el estilo Snake_case.		X	
Los acrónimos técnicos se escribieron todo en mayúscula.		X	
Existe una estandarización de los nombres.		X	
Se mantuvo la indentación de 2 espacios.		X	
Se rompieron líneas de código muy extensas.		X	
Se utilizó la agrupación e indentación utilizando paréntesis.		X	
Existen comentarios innecesarios.		X	
Existen comentarios de código que ha sido eliminado.		X	
Relación lógica del código con relación a la documentación.		X	

## Ast.mli

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Dispone un encabezado para las funciones y lo que estas realizan.	X		
El nombre de las funciones se establece mediante el estilo snake_case.	X		
El nombre de las variables e identificadores se establecen mediante el estilo snake_case.	X		
El nombre de los Constructores se establece mediante el estilo Snake_case.	X		
Los acrónimos técnicos se escribieron todo en mayúscula.		X	
Existe una estandarización de los nombres.	X		
Se mantuvo la indentación de 2 espacios.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.	X		
Existen comentarios innecesarios.		X	
Existen comentarios de código que ha sido eliminado.		X	
Relación lógica del código con relación a la documentación.	X		

## ErrorReporter.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Dispone un encabezado para las funciones y lo que estas realizan.	X		
El nombre de las funciones se establece mediante el estilo snake_case.	X		
El nombre de las variables e identificadores se establecen mediante el estilo snake_case.	X		
El nombre de los Constructores se establece mediante el estilo Snake_case.	X		
Los acrónimos técnicos se escribieron todo en mayúscula.	X		Existe un error en la definición de la función write_x_m_l_errors.
Existe una estandarización de los nombres.	X		
Se mantuvo la indentación de 2 espacios.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.	X		
Existen comentarios innecesarios.		X	
Existen comentarios de código que ha sido eliminado.		X	
Relación lógica del código con relación a la documentación.	X		

## IdentificationTablePrinter\_XML.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Dispone un encabezado para las funciones y lo que estas realizan.	X		
El nombre de las funciones se establece mediante el estilo snake_case.	X		
El nombre de las variables e identificadores se establecen mediante el estilo snake_case.	X		
El nombre de los Constructores se establece mediante el estilo Snake_case.		X	
Los acrónimos técnicos se escribieron todo en mayúscula.		X	
Existe una estandarización de los nombres.	X		
Se mantuvo la indentación de 2 espacios.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.		X	
Existen comentarios innecesarios.		X	
Existen comentarios de código que ha sido eliminado.		X	
Relación lógica del código con relación a la documentación.	X		

## TreeDrawer.ml

Fecha: 21 de enero de 2019.

Nombre del revisor: Gabriel Ramírez Ramírez.

Aspecto de revisión	Presente	No presente	Observaciones
Dispone un encabezado para las funciones y lo que estas realizan.	X		
El nombre de las funciones se establece mediante el estilo snake_case.	X		
El nombre de las variables e identificadores se establecen mediante el estilo snake_case.	X		
El nombre de los Constructores se establece mediante el estilo Snake_case.	X		
Los acrónimos técnicos se escribieron todo en mayúscula.	X		Existe un error en la definición de la función write_x_m_l_tree.
Existe una estandarización de los nombres.	X		
Se mantuvo la indentación de 2 espacios.	X		
Se rompieron líneas de código muy extensas.	X		
Se utilizó la agrupación e indentación utilizando paréntesis.	X		
Existen comentarios innecesarios.		X	
Existen comentarios de código que ha sido eliminado.		X	
Relación lógica del código con relación a la documentación.	X		