**TECNOLÓGICO DE COSTA RICA**

**Escuela de Ingeniería en Computación**

**Proyecto de Ingeniería de Software**

**Profesora:**

María Estrada Sánchez

**Entrega 3:**

Analizador Contextual:

Diseño de pruebas y revisiones.

**Estudiantes:**

Christian León Guevara - 2013371982

Gabriel Ramírez Ramírez - 201020244

**Fecha de entrega:**

20-01-2019

**Período Verano**

**Cartago**

**Tabla de contenido.**

[**Introducción**. 3](#_Toc535919411)

[**Listado de Casos de Prueba.** 3](#_Toc535919412)

[**Revisiones de estilos del código mediante listas de cotejo.** 6](#_Toc535919413)

[ Lista de cotejo de Legibilidad. 6](#_Toc535919414)

[ Lista de cotejo de Comprensibilidad. 7](#_Toc535919415)

# **Introducción**.

En este documento se hace un listado de los casos de pruebas creados para el Analizador Contextual del Lenguaje Triángulo. También se hace una descripción breve del objetivo que cada prueba tiene y el resultado esperado luego de su ejecución.

También se establecieron listas de cotejo para la validación del código desarrollado en Ocaml, dichas listas evaluarán legibilidad, comprensibilidad y estándar de estilos del código

# **Listado de Casos de Prueba.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso** | **Archivo de prueba** | **Objetivo** | **Resultado esperado** |
| 1 | arrAccContextualErrorModoGet.tri | Presenta error contextual cantidad de parámetros en el get. | Incorrecto |
| 2 | arrAccContextualErrorModoPutint.tri | Presenta error contextual cantidad de parámetros en putint. | Incorrecto |
| 3 | arrAccContextualErrorParamsGet.tri | Presenta error contextual cantidad de parámetros en el get. | Incorrecto |
| 4 | arrAccContextualTipoPutInt.tri | Presenta error contextual en tipo de argumento enviado a putint. | Incorrecto |
| 5 | arrAccContextualOKEjecMal.tri | Sin error contextual se prueba el correcto uso del get. | Correcto |
| 6 | constErr01.tri | Presenta error contextual al tratar de cambiar constante | Incorrecto |
| 7 | constErrDeclarationRepeat01.tri | Presenta error contextual al hacer doble declaración de un mismo identificador al mismo nivel. | Incorrecto |
| 8 | constErrMayusYMinus01.tri | Presenta error contextual hay diferencia entre mayúscula y minúscula. | Incorrecto |
| 9 | dirAccDeclaration01.tri | Sin error, se hace un uso correcto de los directorios. | Correcto |
| 10 | dirAccErr01.tri | Error al verificar el tipo correcto del elemento del directorio. | Incorrecto |
| 11 | expressionErr01.tri | Error al revisar que una expresión sea de tipo correcto. | Incorrecto |
| 12 | funcDeclaration01.tri | Error al chequear de parámetros con nombres repetidos. | Incorrecto |
| 13 | funcDeclaration02.tri | Sin error al revisar el proc y func. | Correcto |
| 14 | IfErrExpression01.tri | Error al verificar que la expresión más a la izquierda del if sea booleana. | Incorrecto |
| 15 | IfErrExpression02.tri | Sin error al verificar el uso de if como una expresión booleana. | Correcto |
| 16 | IfErrExpression03.tri | Sin error al verificar el uso de if como una expresión booleana. | Correcto |
| 17 | IfErrExpression04.tri | Error al verificar el uso de if como una expresión booleana. | Incorrecto |
| 18 | IfErrExpression05.tri | Error al verificar el uso de if como una expresión booleana. En los 2 lados del if no concuerdan los tipos de expresión. | Incorrecto |
| 19 | noDeclaractionErr01.tri | Error al existir una variable no definida a la cual se le hace una asignación. | Incorrecto |
| 20 | noDeclaractionErr02.tri | Error al usar de una variable en otro bloque de código diferente. | Incorrecto |
| 21 | procErrConstParamVar01.tri | Error: Un proc llamado con argumento const para un parámetro var | Incorrecto |
| 22 | procErrConstParamVar02.tri | Error: Un proc llamado con argumento const para un parámetro var | Incorrecto |
| 23 | procErrConstParamVar03.tri | Error: Un proc llamado con argumento const para un parámetro var | Incorrecto |
| 24 | procErrDesconocido.tri | Error al llamar un proc desconocido (no declarado). | Incorrecto |
| 25 | procErrFuncParam01.tri | Error al llama un proc con argumento tipo func para un parámetro Integer | Incorrecto |
| 26 | procErrFuncParam02.tri | Error al llama un proc con argumento tipo func para un parámetro Integer | Incorrecto |
| 27 | procErrNumParams01.tri | Error el proc recibe demasiados parámetros. | Incorrecto |
| 28 | procErrNumParams02.tri | Error el proc recibe pocos parámetros. | Incorrecto |
| 29 | procErrParamInvisible.tri | Error al no declarar una variable fuera de un proc. | Incorrecto |
| 30 | procErrParamsRepet01.tri | En una misma abstracción (función, procedimiento), cada parámetro debe aparecer una sola vez en la secuencia. | Incorrecto |
| 31 | procErrProcParam01.tri | Error de un proc que llama a otro proc sin embargo el tipo debe ser Integer. | Incorrecto |
| 32 | procErrProcParam02.tri | Error de un proc que llama a otro proc sin embargo el tipo debe ser Integer. Los procedimientos no retornan valores. | Incorrecto |
| 33 | procErrTipoParam.tri | Error al un llamar proc con parámetro de tipo equivocado. | Incorrecto |
| 34 | procErrTipoParam01.tri | Error ya que un tipo no puede usarse como argumento en ninguna circunstancia | Incorrecto |
| 35 | procOKFuncParam01.tri | Proc llamado con argumento func que retorna un Integer para parámetro Integer. | Correcto |
| 36 | procOKParam.tri | Procedimiento bien en cantidad de parámetros, tipo y modo. | Correcto |
| 37 | varAssign.tri | Prueba para la asignación y validación de tipos. | Correcto |
| 38 | varErrAssign.tri | Error en la prueba para la asignación y validación de tipos. Diferencia de tipo. | Incorrecto |
| 39 | whileErrExpression01.tri | Error en la prueba para la validación de tipos. La expresión del while tiene ser boolean. | Incorrecto |
| 40 | whileErrExpression02.tri | Error en la prueba para la validación de tipos. La expresión del while tiene ser boolean. | Incorrecto |

# **Revisiones de estilos del código mediante listas de cotejo.**

* Lista de cotejo de Legibilidad.

Ejecución de la Legibilidad:

Fecha:

Nombre del documento:

Nombre del revisor:

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

* Lista de cotejo de Comprensibilidad.

Ejecución de la Comprensibilidad:

Fecha:

Nombre del documento:

Nombre del revisor:

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