**TECNOLÓGICO DE COSTA RICA**

**Escuela de Ingeniería en Computación**

**Proyecto de Ingeniería de Software**

**Profesora:**

María Estrada Sánchez

**Entrega 3:**

Analizador Contextual:

Catálogo de los casos de prueba.

**Estudiantes:**

Christian León Guevara - 2013371982

Gabriel Ramírez Ramírez - 201020244

**Fecha de entrega:**

20-01-2019

**Período Verano**

**Cartago**

**Tabla de contenido.**

[**Introducción**. 2](#_Toc535743580)

[**Listado de Casos de Prueba.** 2](#_Toc535743581)

[**Revisiones de estilos del código mediante listas de cotejo.** 3](#_Toc535743582)

[Lista de cotejo de Legibilidad. 3](#_Toc535743583)

[Lista de cotejo de Comprensibilidad. 3](#_Toc535743584)

[**Revisiones estáticas del código.** 3](#_Toc535743585)

[OcamlLint. 3](#_Toc535743586)

[OUnit. 3](#_Toc535743587)

# **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.

# **Listado de Casos de Prueba.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Archivo de prueba** | **Encargado** | **Objetivo** | **Resultado obtenido** | **Resultado esperado** |
| caso1.tri | Gabriel Ramírez | Validar la asignación y tipos de declaraciones. | Correcto | Correcto |
| caso2.tri | Gabriel Ramírez | Validar la asignación y tipos de declaraciones. | Incorrecto | Incorrecto |
| caso3.tri | Gabriel Ramírez | Validación del Tipo Entero. | Correcto | Correcto |
| caso4.tri | Gabriel Ramírez | Validación del Tipo Entero. | Incorrecto | Incorrecto |
| caso5.tri | Gabriel Ramírez | Validación del Tipo Char. | Correcto | Correcto |
| caso6.tri | Gabriel Ramírez | Validación del Tipo Char. | Incorrecto | Incorrecto |
| caso7.tri | Gabriel Ramírez | Validación del Tipo Boolean. | Correcto | Correcto |
| caso8.tri | Gabriel Ramírez | Validación del Tipo Boolean. | Incorrecto | 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. |  |  |  |

# **Revisiones estáticas del código.**

* OcamlLint.

Se usará está herramienta para probar el código desarrollado y se hará un análisis de resultados con la evidencia obtenida.

* OUnit.
* Se usará está herramienta para probar el código desarrollado y se hará un análisis de resultados con la evidencia obtenida.