

## Versión [1.0]



### Historia de revisiones

Fecha	Versión	Descripción
[09/04/2019]	[1.0]	[Integers]

# Contenido

1. INTRODUCCIÓN..... ¡ERROR! MARCADOR NO DEFINIDO.
  - 1.1. PROPOSITO, ALCANCE Y OBJETIVOS..... ¡ERROR! MARCADOR NO DEFINIDO.
  - 1.2. RESPONSABLES DEL PROYECTO..... ¡ERROR! MARCADOR NO DEFINIDO.
2. FUNCIONALIDAD DEL PRODUCTO¡ERROR! MARCADOR NO DEFINIDO.
  - 2.1. MODULOS FUNCIONALES DEL PRODUCTO ¡ERROR! MARCADOR NO DEFINIDO.
  - 2.2. FUNCIONES ESPECIFICAS DEL PRODUCTO ¡ERROR! MARCADOR NO DEFINIDO.
3. PRERREQUISITOS DE USO DEL PRODUCTO¡ERROR! MARCADOR NO DEFINIDO.
  - 3.1. OBJETIVOS Y PRIORIDADES DE GESTIÓN ... ¡ERROR! MARCADOR NO DEFINIDO.
  - 3.5. RECURSOS..... ¡ERROR! MARCADOR NO DEFINIDO.
4. ELEMENTOS ENTREGABLES DEL PROYECTO¡ERROR! MARCADOR NO DEFINIDO.
  - 4.1. PROCEDIMIENTOS TÉCNICOS, HERRAMIENTAS Y TECNOLOGÍAS .....¡ERROR! MARCADOR NO DEFINIDO.
  - 4.2. DOCUMENTACIÓN DE SOFTWARE ..... ¡ERROR! MARCADOR NO DEFINIDO.

## 1. INTRODUCCIÓN

### 1.1. Propósito, Alcance y Objetivos del Proyecto

El propósito de este documento es analizar los aspectos relevantes, que se tendrán en cuenta durante las etapas de diseño, implementación e instalación del producto, teniendo en cuenta las necesidades mencionadas por los interesados de la entidad. Se pretende diseñar, implementar, probar e instalar un sistema capaz de leer, analizar, validar y generar el ejecutable del subconjunto de c especificado, Dado una entrada -> genera un ejecutable.

Se espera entregar un producto en una versión estable, que satisfaga los requerimientos acordados, para ser puesto en producción lo antes posible debido a la necesidad actual, después de haber pasado por una etapa de pruebas.

### 1.2. Responsables

Se identifican las actividades más relevantes en el proyecto, los responsables de dichas actividades y los involucrados.

Identificación de actividad	Descripción de actividad	Responsable
Project Manager	Responsable de monitorear el cumplimiento del Plan de Proyecto	Luis Martin Maceda Nazario
System Architect	Responsable del desarrollo de la arquitectura utilizada para el software	Gómez Trejo Gustavo Ali
System Tester	Responsable de pruebas automatizadas para comprobar el correcto funcionamiento del software	Hernández Albino Edgar Alejandro
Integrator	Responsable del	Rangel García Carlos Alberto

## 2. FUNCIONALIDADES DEL PRODUCTO

### 2.1 Módulos Funcionales del producto

ID	Nombre	Descripción
RF	Ejecutable	
RF-1	Scanner (lexer)	Es la primera fase de un compilador consistente en un programa que recibe como entrada el código fuente de otro programa (secuencia de caracteres) y produce una salida compuesta de <i>tokens</i> (componentes léxicos) o símbolos.
RF-1.1	<u>Sanitizer</u>	Es un módulo externo al Scanner y su principal función es el saneamiento de código para exceso de espacios, tabulaciones, etc.
RF-2	Parser	Es un programa informático que analiza una cadena de símbolos de acuerdo a las reglas de una gramática formal. El análisis sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada.
RF-2.1	AST	Un árbol de sintaxis abstracta (AST), o simplemente un árbol de sintaxis, es una representación de árbol de la estructura sintáctica simplificada del código fuente escrito en cierto lenguaje de programación. Cada nodo del árbol denota una construcción que ocurre en el código fuente.
RF-3	Generator	convierte parte representación intermedia de código fuente en una forma (por ejemplo, el código máquina ) que puede ser fácilmente ejecutado por una máquina
RF-3.1	Linker	Es un módulo externo al Generator y su principal función es ser la conexión entre el archivo ensamblador generado y el gcc del sistema operativo, para poder generar el ejecutable correspondiente del archivo .C.

## 2.2 Funciones Específicas del producto

- Lista de Tokens: Es la lista de elementos tokens) que se obtiene después de realizar el análisis léxico (Scanner, sanitizer), esta lista debe poder ser mostrada en consola si el usuario así lo desea.
- Árbol AST: Es el árbol de sintaxis abstracta el cual se genera de forma automática durante proceso de compilación (Parser, AST) para poder identificar que el orden elementos/símbolos que estén dentro del archivo .c, es árbol debe poder ser mostrado en consola si el usuario así lo desea.
- Código Ensamblador: Archivo que contiene el código ensamblador procesado del archivo .c que se introduce para ser compilado.
- Banderas: utilizando el estándar de UNIX (En esta primera versión solo una bandera).
  - Opción -o nombre\_ejecutable (escoger el nombre del ejecutable)
  - Opción -s ensamblador (genera el archivo con código ensamblador resultante del análisis léxico y procesamiento sintáctico del archivo .C, no genera ejecutable)
  - Opción -t Tokens (Muestra la lista de tokens generada (Scanner, sanitizer), No genera ejecutable)
  - Opción -a AST (Muestra el árbol AST, No genera ejecutable)
  - Opción -h help (instrucciones para el uso del compilador)

### 3. PRERREQUISITOS DE USO DEL PRODUCTO

#### 3.1. Objetivos y Prioridades de Gestión

- Sistema Operativo: Para poder utilizar el producto es necesario contar con un sistema operativo tipo UNIX.
- Elixir: Tener instalado Elixir dentro de los dispositivos donde se desee utilizar el producto.
- Gcc: Tener instalado el conjunto de compiladores creados por el proyecto GNU.

### 4. ELEMENTOS ENTREGABLES DEL PROYECTO

Identificación Entregable	Descripción Entregable	Fecha de entrega	Lugar de entrega	Condiciones satisfacción
L.R.	Levantamiento de requerimientos	09/04/2019	F. I. Edificio B. Aula 205	
ARQ.	Arquitectura de Proyecto	09/04/2019	F. I. Edificio B. Aula 205	
P. T	Plan de Trabajo	09/04/2019	F. I. Edificio B. Aula 205	
Manual	Manual de uso del producto	09/04/2019	F. I. Edificio B. Aula 205	