**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería**

**Especificaciones de proyecto para el curso de compiladores.**

**Fecha: 9- abril- 2019**



# Índice

1. [Introducción 2](#_TOC_250014)
   1. [Propósito 2](#_TOC_250013)
   2. [Alcance 2](#_TOC_250012)
   3. [Abreviaturas 2](#_TOC_250011)
   4. [Apreciación global 3](#_TOC_250010)
2. [Descripción global 3](#_TOC_250009)
   1. [Perspectiva del producto 4](#_TOC_250008)
   2. [Funciones del producto 4](#_TOC_250007)
   3. [Características de los usuarios 4](#_TOC_250006)
   4. Supuestos y referencias 5
3. [Requerimientos 5](#_TOC_250004)
   1. [Requerimientos funcionales 5](#_TOC_250003)
   2. [Requerimientos no funcionales 6](#_TOC_250002)
4. [Anexos 7](#_TOC_250001)
   1. [Directorio 7](#_TOC_250000)

# Introducción

El equipo de desarrollo “ALFA” busca poner en práctica los conocimientos adquiridos hasta el momento del curso de compiladores, debido a esto se realizará este proyecto el cual consta en la implementación de un compilador que tendrá un funcionamiento básico ya que de forma general deberá leer texto con la sintaxis perteneciente al lenguaje de programación “C” y por lo consiguiente generar un archivo ejecutable.

Este proyecto será dirigido tanto para los involucrados en el mismo como para quienes se encuentren dentro del campo de desarrollo de software y busquen conocer lo básico del funcionamiento e implementación de un compilador.

## Propósito

Con el objetivo de alcanzar la mayor calidad del software y tener una mejor visualización de lo que se deberá entregar al finalizar el proyecto se hace uso del presente documento para el análisis del mismo, recopilando los requerimientos funcionales y no funcionales del compilador a desarrollar.

## Alcance

El presente documento contiene los requerimientos tomados, así como perfiles que tendrán los usuarios, aquellos que son tanto administrativos como clientes. Aquí no se tomará la parte de administración de proyectos, contrato de personal, contratos legales o diseño de la solución.

## Abreviaturas

|  |  |
| --- | --- |
| Abreviatura | Significado |
| ISA | Intruction Set Arquitecture (conjunto de instrucciones de arquitectura) |
| ERS | Especificaciones de Requerimientos de Software |
| SO | Sistema Operativo |
| X86 | Familia de procesadores de Intel |
| AST | Abtract Sintaxis Tree (Árbol de sintaxis abstracta) |
| C | Lenguaje de programación |
| UNIX | Sistema Operativo |

## 1.4 Apreciación global

El ERS muestra de lo general a lo particular el problema y la solución para desarrollar este proyecto describiendo el funcionamiento esperado, desde los requerimientos funcionales hasta los no funcionales que deben de ser considerados en el diseño del software.

# Descripción global

El equipo de desarrollo “Alfa” busca poner en práctica los conocimientos que se han adquirido hasta este momento del curso de compiladores a través de la implementación de un compilador que será desarrollado en el lenguaje de programación funcional “Elixir” el cual tendrá la capacidad de leer texto que se encuentre con la sintaxis perteneciente al lenguaje de programación “C” y como consecuencia generará un archivo ejecutable y en el caso de que el texto leído no este escrito acorde a la sintaxis establecida se hará saber al usuario del error existente.

Este proyecto estará dirigido tanto para los involucrados en el mismo como también a las personas que estén interesadas en el área de desarrollo de software y que quieran conocer el funcionamiento básico de un compilador, por lo tanto todo el contenido de este proyecto estará disponible en su totalidad en un repositorio de GitHub para así tener un mejor control del proyecto y sus versiones las cuales estarán a cargo de los integrantes del equipo de desarrollo además los responsables de dar o no acceso al repositorio a personas ajenas que requieran del contenido.

Las diferentes versiones del proyecto serán presentadas con respecto a las siguientes fechas mostradas las cuales podrán ser modificadas y actualizadas en caso de ser necesario.

**Versión 1: Enteros (Primer demo, Abril 9)**

**Versión 2: Operadores unarios (Segunda demo, Mayo 14)**

**Versión 3: Operadores Binarios (Tercer demo, Junio 7)**

**2.1 Perspectiva del producto**

El compilador será un producto que permitirá ver todas las fases de su ejecución, capaz de leer código y generar el ejecutable, que se pueda correr desde la línea de comandos, tener banderas que ayuden a la impresión de las salidas, poder correr pruebas automatizadas del lenguaje de programación C, sólo es capaz de soportar enteros, operadores unarios y operadores binarios

El producto no contempla actualmente:

* Que se ejecuten pruebas en C, con otra función que no sea main
* Que se ejecute en otro lugar que no sea la línea de comandos
* Correr programas en C, que no sean como los establecidos por el cliente
* Cualquier otro requerimiento que no se especifique explícitamente en el presente

**2.2 Funciones del producto**

|  |  |  |
| --- | --- | --- |
| **ID** | **Nombre** | **Descripción** |
| **RF1** | **Capaz de leer código** | **El código tendrá que tener cierto formato para que lo pueda leer el compilador** |
| **RF2** | **Imprimir la salida del lexer** | **El sistema será capaz de imprimir la lista de tokens que recibe del programa y confirmar que son elementos del lenguaje.** |
| **RF3** | **Imprimir la salida del parser** | **El sistema checará la sintaxis, si todo está correcto, convertirá la salida en una estructura de árbol (AST) y lo imprimirá en pantalla.** |
| **RF4** | **Generar el ensamblador** | **Al escribir una bandera, el sistema será capaz de mostrar el ensamblador** |
| **RF5** | Mostrar ayuda | **El sistema será capaz de mostrar la ayuda al ejecutar una bandera** |

## Características de los Usuarios

## Este compilador podrá ser manejado/utilizado por personas que manejen lo básico de línea de comandos en UNIX.

## Si sabes utilizar línea de comandos pero no sabes lo que es un compilador y sus fases, te recomendamos investigar para que puedas entender mejor lo que te estamos mostrando.

## También comprender las funciones de los operadores binarios, como lo es la negación, el complemento, y la negación lógica.

## Supuestos y dependencias

Se espera que al finalizar el proyecto tanto los involucrados y otros usuarios puedan hacer total uso del compilador. Se considera que es fundamental la capacitación de los integrantes del equipo para conocer los conceptos necesarios para poder implementar el compilador. Se considera que se nos brinden los permisos necesarios en GitHub para poder hacer uso de esta herramienta durante toda la realización del proyecto.

# Requerimientos

## Requerimientos Funcionales

|  |  |
| --- | --- |
| **NOMBRE** | **DESCRIPCIÓN** |
| Lectura de texto | El compilador será capaz de leer texto que será tomando en cuenta la sintaxis de C |
| Generar ejecutable | Después de haber leído el texto como resultado se generará un archivo ejecutable |
| Reporte de fallos | En caso de haber algún error de sintaxis en el texto leído se le reportara al usuario indicando la línea en la que se encuentra el error |
| Ubicación y nombre del ejecutable | Por default el ejecutable tendrá el mismo directorio y el mismo nombre que el archivo del cual se leyó el texto |
| Bandera O | Esta bandera dará la oportunidad de al usuario de agregarle el nombre deseado al ejecutable |
| Bandera S / ensamblador | Esta bandera hará que el compilador solo devuelva el archivo con el código ensamblador generado |
| Bandera T / tokens | Esta bandera hará que el compilador solo muestre en la consola la lista con los tokens obtenidos del archivo leído |
| Bandera A / ast | Esta bandera permitirá al usuario visualizar solamente el árbol AST en consola |
| Bandera H / help | Esta bandera le mostrara al usuario instrucciones de como correr el compilador |
| Pruebas automatizadas | El compilador será capaz de realizar pruebas automatizadas |

## Requerimientos no funcionales

|  |  |
| --- | --- |
| **NOMBRE** | **DESCRIPCIÓN** |
| Sistema operativo | El sistema operativo en el cual funcionará el compilador será en UNIX |
| Manejo de Git | El manejo del contenido del proyecto será por medio de la herramienta de control de versiones Git |
| Consola de comandos | Para poder probar el funcionamiento del compilador se deberá hacer uso exclusivo de la consola de comandos |
| Lenguaje Elixir | El compilador será desarrollado en el lenguaje de programación funcional Elixir |
| Comandos | Se deberán hacer uso de ciertos comandos para poder realizar las pruebas del compilador |

# Anexos

## Directorio

|  |  |
| --- | --- |
| **Cargo** | **Información** |
| Stakeholder (interesado) | Nombre: Norberto Ortigoza |
| Lider del Proyecto | Nombre: Norberto Ortigoza |
| Tester | Nombre: Islas Martínez Kevin Yael |
| Integrador | Nombre: Ramírez Fierro Salma Arelly |
| Arquitecto | Nombre: Corona Vera Logan Alejandro |
| Project Manager y Desarrolladora | Nombre: Sánchez Castillo Paola Elizabeth |