Objetivos

Diseñar un compilador en C++ que permita transformar desde un lenguaje fuente a un lenguaje objeto, en nuestro caso el lenguaje “alfanumérico” implementando una tabla de símbolo que nos permita la traducción de un lenguaje de programación (C++) por un compilador o un intérprete (C++), donde cada símbolo en el código fuente de un programa está asociado con información tal como el tipo de datos, ordenamiento de letras alfabéticamente y suma de los valores numéricos leídos.

Lenguaje

**Ventajas:**

- Lenguaje de programación orientado a objetos.   
- Lenguaje muy didáctico, gracias a este lenguaje puedes aprender muchos otros lenguajes con gran facilidad, como C#, Java, Visual Basic, Javascript, PHP, entre otros.   
- Es muy potente en lo que se refiere a creación de sistemas complejos, un lenguaje muy robusto.   
- Permite elaborar aplicaciones sencillas, hasta sistemas operativos y mucho más, todo eso dependiendo del manejo del lenguaje.   
- Actualmente, puede compilar y ejecutar código de C, ya viene con librerías para realizar esta labor.   
- Es un lenguaje muy empleado, existen muchos tutoriales en línea, libros, códigos fuentes abiertos... hay material de sobra y basta para aprender lo necesario y mucho más con este lenguaje.   
- Existen muchos algoritmos cuyo pseudocódigo se encuentra ya desarrollado en C++, de manera que puedes tomarlo y amoldarlo a tu solución (porque el que veas un fragmento de código no asegura que sea correcto al 100%).

**Desventajas:**

- Uso de DLLs (librerías dinámicas) muy complejo. Java y .Net han evolucionado estos conceptos manipulando las DLLs mediante los frameworks que proveen. En cambio, en C++ el desarrollador debe encargarse de cargar y liberar de memoria estas librerías, y correr los riesgos por el manejo de esta memoria.   
- Elaborar un sistema en C++ es como construir un rascacielos: tiene buen soporte y es robusto, pero si existen errores en los pisos inferiores toda la parte superior se viene abajo terriblemente.   
- Manejo de punteros y memoria respecto a ello. Claro, esta también es una gran ventaja porque permite un mejor control de la memoria y una buena administración de recursos de computadora, pero la inexperiencia de los desarrolladores o la pérdida de costumbre con este tipo de variables los lleva al desastre.   
- No es recomendable para desarrollo de páginas Web.

Arquitectura

El programa consta de dos fases: análisis léxico, análisis sintáctico.

En el análisis léxico, el programa reconoce cada carácter de la cadena que se introduce con la ayuda de la tabla de símbolos que tenemos.

En el análisis sintáctico, el programa procesa las caracteres que provienen del análisis léxico y dependiendo si son letras del alfabeto o números.

Descripción de la Utilidad

El compilador tiene la utilidad de leer el código fuente (C++) y transformarlo a alfanumérico además de ordenar las letras en alfabéticamente y sumar los números leídos.

Descripción de la tabla de símbolo

Nuestra tabla de símbolo comprende código fuente C++ y las traduce a letras que van desde a la

(a b c d e f g h i j k l m n o p q r s t u v w x y z) y números de la (0,1,2,3,4,5,6,7,8,9).

# Manejo de errores

El compilador manejo 2 tipos de errores en el programa:

1. Si el usuario introduce una cadena vacía, el compilador manda un mensaje al usuario y le pide reintroducir una cadena valida.
2. Si el usuario introduce algún símbolo que no está presente en la tabla (+,-,\*,!,",#,$,%,etc), el compilador detiene la lectura de la cadena y manda un mensaje de error al usuario (Ejemplo: El caracter '+' no es un caracter válido) y pide al usuario introducir una cadena válida.

**Diagrama T**

