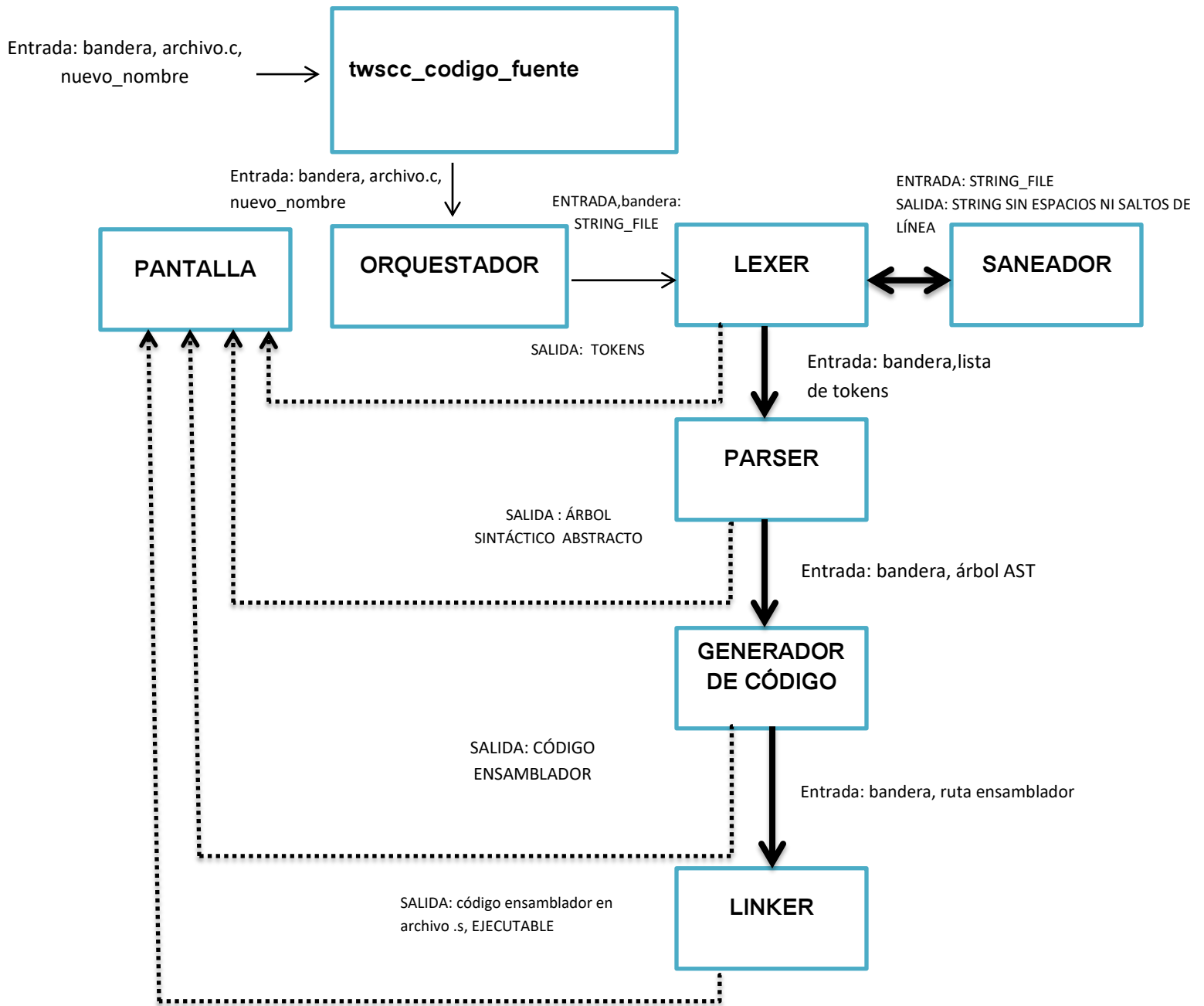


ARQUITECTURA DEL COMPILADOR



SIMBOLOGÍA



ENTRADA DEL MÓDULO



SALIDA DEL MÓDULO OPCIONAL

DEFINICIÓN DE REPSONSABILIDADES DE CADA MÓDULO EN LA ARQUITECTURA

Módulo: twscc_codigo_fuente

Este módulo es el encargado de recibir tres argumentos desde la línea de comandos. El primer argumento hace referencia a las banderas (opciones de compilación) previamente definidas en los requerimientos.

El segundo argumento que recibe es la dirección del archivo con extensión .c que debe compilar el programa.

En el tercer argumento, el usuario tiene la opción de renombrar el archivo .s con el nombre de su elección, previamente se debió haber elegido la bandera -o para acceder a esta opción.

Finalmente se manda a llamar al módulo del Orquestador.

Módulo: Orquestador

El módulo orquestador recibe los tres argumentos de entrada. Su primera responsabilidad consiste en leer el archivo .c y guardarlo para que pueda ser leído por los demás módulos. Además tiene la responsabilidad de coordinar el llamado de los demás módulos funcionales del programa y pasar la bandera que indica la impresión en pantalla del resultado de cada módulo. Para lograr una coordinación completa del programa, se decidió usar pipes para interconectar los módulos, de esta manera la salida de un módulo será la entrada de otro respetando el diagrama de arriba.

Módulo Lexer

Este módulo recibe como entrada el archivo ya leído y la bandera desde el Orquestador a su vez, manda a llamar al módulo Saneador el cual limpiará el código .c. Posteriormente al recibir de regreso el archivo limpio, se hará la división de tokens para introducirlos en una lista y mostrarla en pantalla en caso necesario. En caso de recibir algún parámetro que no está definido en los tokens válidos, lanzará un mensaje de error.

Módulo: Saneador

Este módulo tiene como responsabilidad quitar los saltos de línea o tabuladores del código para limpiarlo y que el Lexer pueda realizar correctamente la división de tokens.

Módulo Parser

Este módulo recibirá la bandera y la lista de tokens desde el Lexer para validar la sintaxis de los mismos y construir el árbol AST que servirá para generar el código ensamblador, en caso de una sintaxis inválida la ejecución del módulo se detiene y lanza un mensaje de error.

Módulo Generador de código

Recibe el árbol AST generado por el Parser y la bandera para hacer un recorrido en PostOrden e ir generando nodo por nodo el código correspondiente en lenguaje ensamblador escribiéndolo en un archivo con extensión .s el cual servirá para generar el ejecutable del programa.

Módulo Linker

Este módulo es el encargado de generar el ejecutable del programa en el mismo directorio en el que se encuentra el archivo .s, recibirá como parámetros la ruta del archivo ensamblador previamente creado y una bandera. Tiene como responsabilidades escribir en el archivo ensamblador el código ensamblador correspondiente previamente creado por el Generador de código y generar el ejecutable.

