UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

LENGUAJES FORMALES Y DE PROGRAMACIÓN

CATEDRÁTICO: ING. DAVID MORALES

TUTOR ACADÉMICO: HERBERTH ABISAI AVILA



**MANUAL TÉCNICO DE PRÁCTICA 1**

ENNER ESAÍ MENDIZABAL CASTRO

CARNÉ: 202302220

SECCIÓN: B+

GUATEMALA, 13 DE AGOSTO DEL 2,024

# ÍNDICE

[ÍNDICE 1](#_Toc174478348)

[INTRODUCCIÓN 2](#_Toc174478349)

[OBJETIVOS 2](#_Toc174478350)

[1. GENERAL 2](#_Toc174478351)

[2. ESPECÍFICOS 2](#_Toc174478352)

[ALCANCES DEL SISTEMA 3](#_Toc174478353)

[ESPECIFICACIÓN TÉCNICA 3](#_Toc174478354)

[● REQUISITOS DE HARDWARE 3](#_Toc174478355)

[● REQUISITOS DE SOFTWARE 3](#_Toc174478356)

[LÓGICA Y DESCRIPCIÓN DE LA SOLUCIÓN 4](#_Toc174478357)

# INTRODUCCIÓN

Este manual se creó con el fin de proporcionar la información técnica, necesario para la comprensión del funcionamiento lógico del programa, esto con un lenguaje más técnico que permita una comprensión más profunda de las funciones, variables y todo lo utilizado para la creación del programa.

# OBJETIVOS

## GENERAL

* 1. Describir el funcionamiento del programa desde una perspectiva más allegada a su lógica y a sus algoritmos.

## ESPECÍFICOS

* 1. Especificar las funciones y procedimientos que se llevaron a cabo para realizar cada una de las funcionalidades del programa
  2. Explicar la manera en la que se resolvieron los problemas que surgieron durante la creación del programa.

# ALCANCES DEL SISTEMA

Este manual se creó con el fin de propiciar una comprensión profunda sobre la manera en la cual se desarrolló la creación del programa y la solución de los problemas durante su elaboración. Se pretende proporcionar un conocimiento técnico tal que permita enseñar su funcionamiento con las suficientes explicaciones para que, si se desea, pueda ser replicado a posterior.

# ESPECIFICACIÓN TÉCNICA

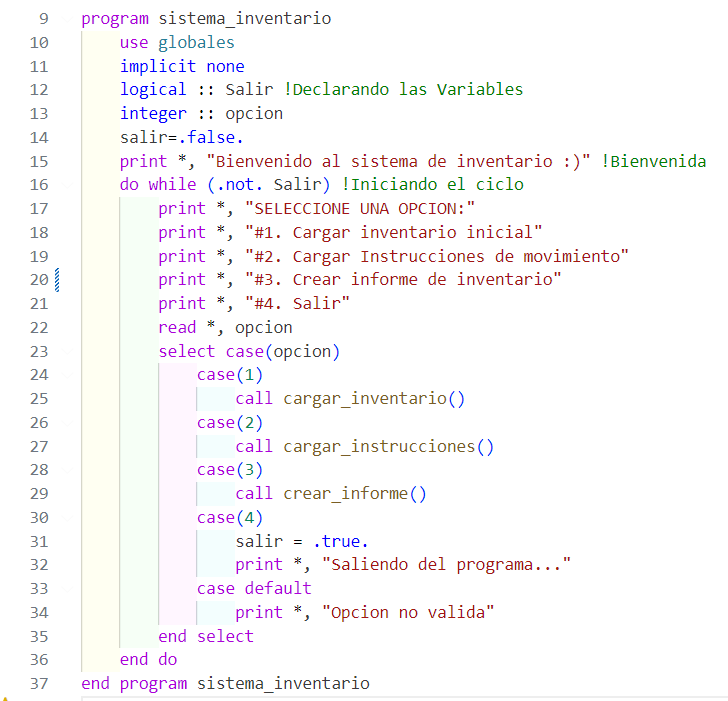
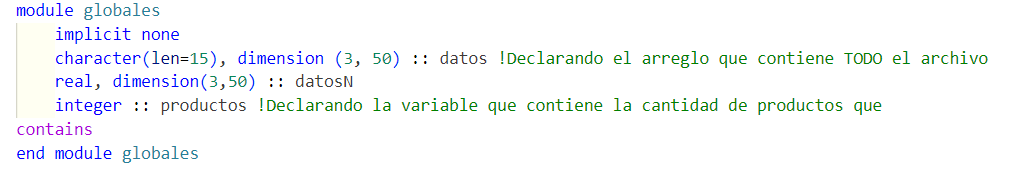
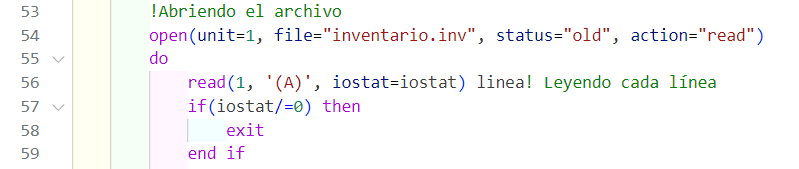
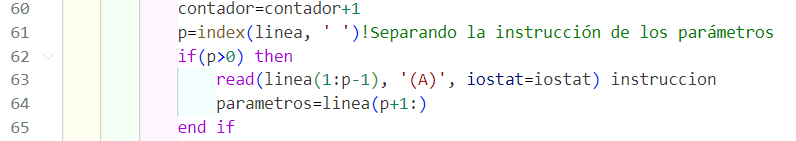
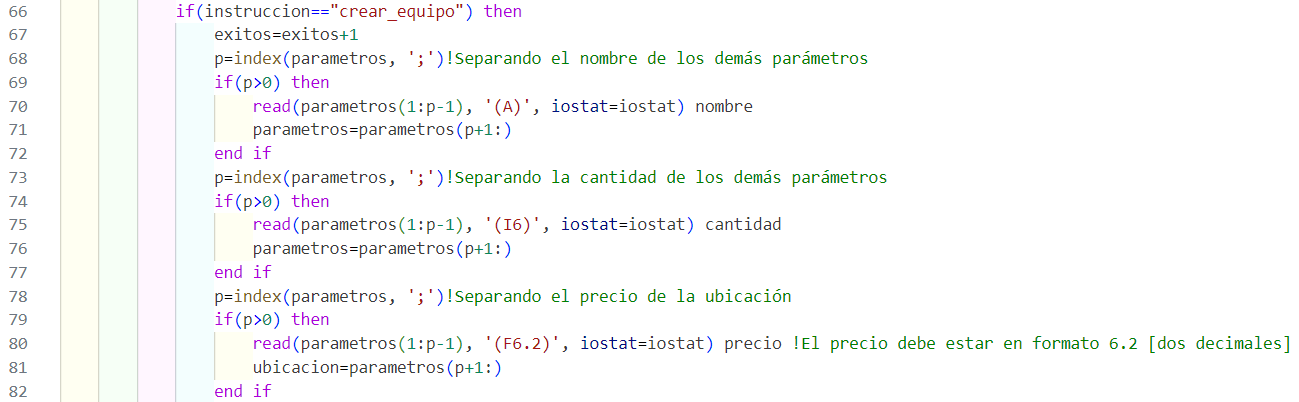
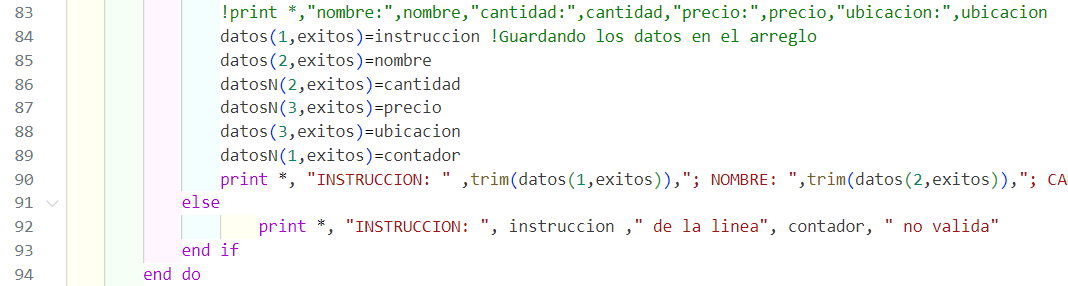
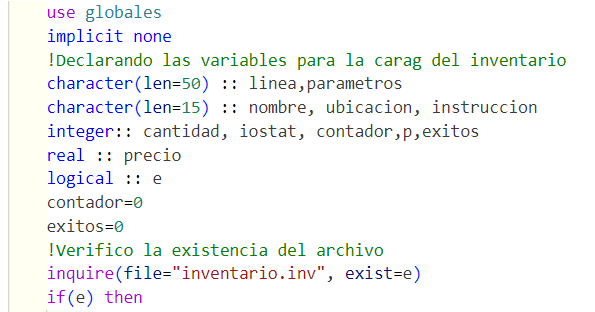
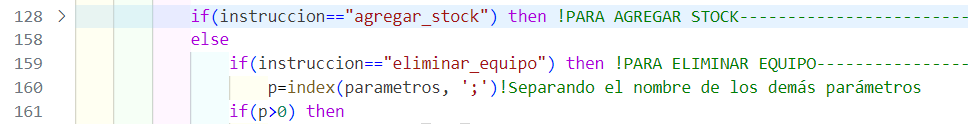
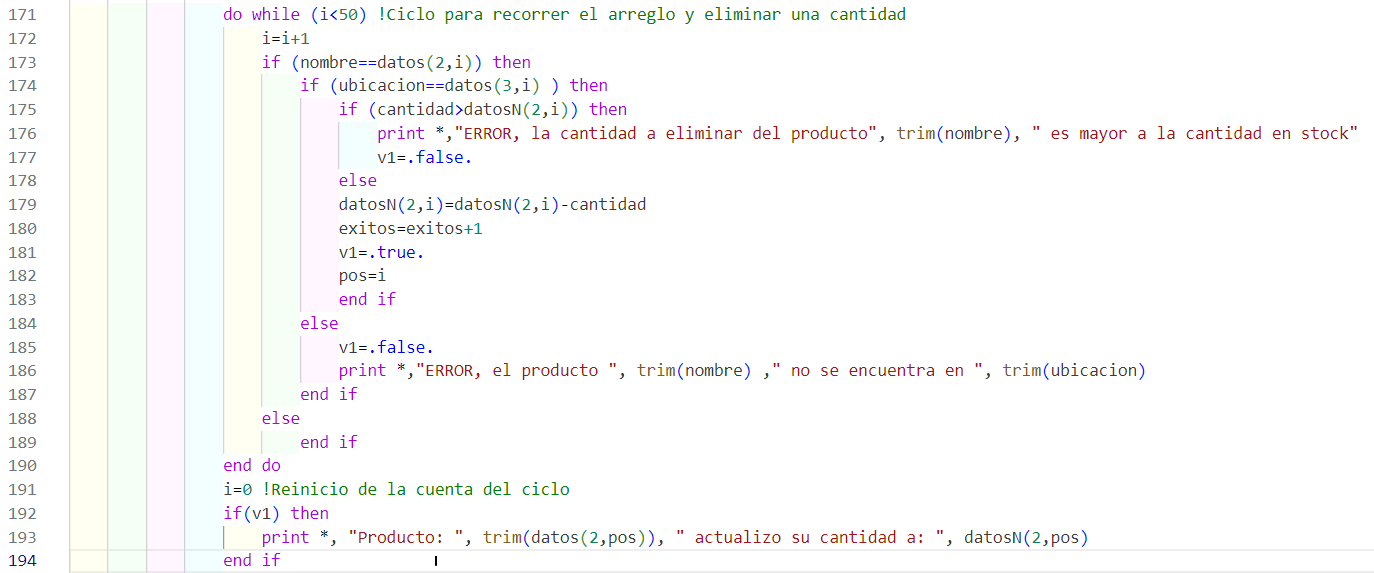
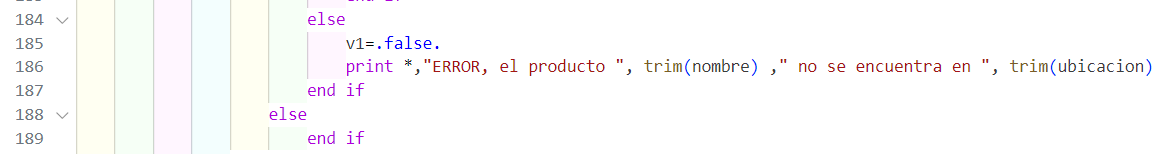
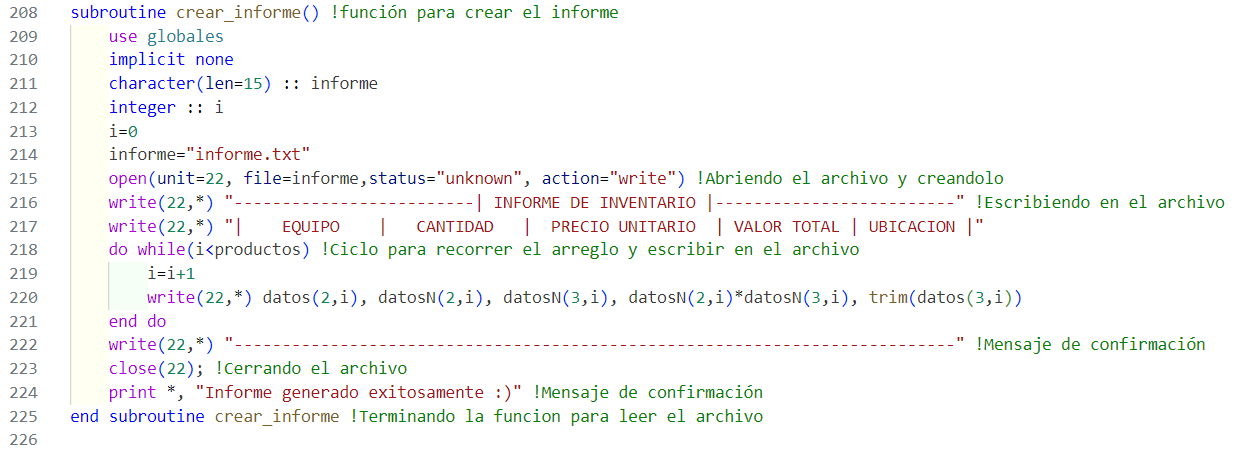
## REQUISITOS DE HARDWARE

* + Procesador con arquitectura x86
  + Teclado (opcional)

## REQUISITOS DE SOFTWARE

* + Editor de texto compatible con Fortran
  + Compilador de Fortran

# LÓGICA Y DESCRIPCIÓN DE LA SOLUCIÓN

* **Creación del menú principal:** Al iniciarse el programa se muestra un menú que permite acceder a cada una de las funciones del programa Para la creación de este, se decidió utilizar un ciclo *while*, del cual no dejaría de estar en ciclo hasta que la variable *booleana* pasara a ser verdadera, lo cual únicamente sucedería si se presionaba el número 4, el cual era para salir. Para acceder a cada unas de las distintas funcionalidades, a excepción de la 4, se crearon distintas subrutinas que se ejecutarían de acuerdo con la opción seleccionada en el switch en ciclado dentro. 
* **Almacenamiento de todos los datos del programa para luego imprimirlos:** Para conseguir almacenar correctamente los datos que se van cargando y modificando durante la ejecución del programa, se decidió utilizar como solución un arreglo estático declarado de forma global, puesto a que no existe los arreglos dinámicos dentro de fortran, en el cual se podrían almacenar todos los datos deseados hasta el límite de cada uno de estos arreglos bidimensionales. Fue creado uno para las variables de tipo *string* y otro para las variables de tipo *real*. 
* **Análisis léxico para la separación de los enunciados:** Para realizar esto, primero se recorrió cada una de las líneas del archivo. Posteriormente, dentro del ciclo mismo que recorre cada una de las líneas del archivo, se separa cada línea en dos a partir de espacio (“ ”) que hay entre la instrucción y los parámetros. Una vez teniendo estos separados, se verifica si la instrucción es valida y, en dado caso esta esté bien declarada, se comienza a separar cada uno de los parámetros y se guardan en variables, así como se hizo al inicio separando las instrucciones de los parámetros. Ya con todo separado dentro de cada unas de sus respectivas variables declaradas previamente, se coloca todo dentro de los dos arreglos bidimensional creados en global para poder guardar los todos estos datos y que puedan ser recorridos y obtenidos nuevamente; y sin en dado caso, la instrucción está mal desde el inicio, indicará que esta está mal y no se reproducirá nada. Este ciclo se copio y pegó en la segunda subrutina (la que se ejecuta al presionar el 2) adaptándola a la cantidad de variables.
* **Cargar Instrucciones:** Esto se realiza mediante una subrutina la cual utiliza el procedimiento de la separación de los enunciados para obtener los parámetros del archivo *inventario.inv* el cual se abre previamente y se le asigna un *unit* con el cual se identificará este archivo. Luego se guarda todo en los arreglos globales y se va mostrando información en la consola conforme se va realizando cada análisis sintáctico. 
* **Cargar instrucciones:** Para esta, como se menciona anteriormente, se toma la lógica para la separación de los enunciados y se separa todo, así como se mostro nuevamente, adaptándolo a la cantidad de variables para esta subrutina y colocando un condicional adicional para verificar si la instrucción es para eliminar o para adicionar productos. Una vez realizado todo esto, en vez de colocar todo dentro de los arreglos globales, se modifican primeramente buscándolos dentro de estos arreglos y luego colocándolos o eliminándolos en la posición encontrada para así poder luego confirmar que se pudo realizar la operación deseada con el producto deseado. Adicionalmente, si se encuentra el producto, mas no se encuentra en la ubicación especificada, indicará que no está donde se indica y no se realizará ningun procedimiento. 
* **Creación de informe:** Para realizar el informe, primero se crea un archivo con el nombre de *informe.txt* (en dado caso ya exista, se sobrescribirá este mismo ya existente), luego se abrirá para poder escribir en él mediante un ciclo el cual repasa cada uno de los valores almacenados dentro de los arreglos globales y los escribe en el con el formato deseado. Al finalizar indica en la consola que se generó exitosamente el informe. 
* **Salir del programa:** Para esto simplemente se cambia el valor de la variable de tipo booleano que mantiene el programa en ciclado, para que este deje estarlo y se ejecute hasta acabarse. Además, si no se escogió ninguna de las opciones validas, se indicará que no se ingreso la una opción válida. 