

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
LENGUAJES FORMALES Y DE PROGRAMACION  
CATEDRÁTICO: ING. OTTO AMILCAR RODRIGUEZ  
TUTOR ACADÉMICO: GERSON QUIROA



## **MANUAL TÉCNICO**

Brayan Emanuel Garcia

CARNÉ: 202300848

SECCIÓN: A

GUATEMALA, 22 DE AGOSTO DEL 2,024

# ÍNDICE

## Contenido

ÍNDICE .....	1
INTRODUCCIÓN .....	2
OBJETIVOS .....	2
1.    GENERAL .....	2
2.    ESPECÍFICOS .....	2
ALCANCES DEL SISTEMA.....	3
ESPECIFICACIÓN TÉCNICA.....	3
•    REQUISITOS DE HARDWARE .....	3
•    REQUISITOS DE SOFTWARE .....	3
ARQUITECTURA DE SOFTWARE .....	4
LÓGICA DEL PROGRAMA .....	5
•    MODULO DE GESTORINVENTARIO:.....	5
•    MAIN.f90: .....	8

# **INTRODUCCIÓN**

Bienvenido al Manual Técnico del Sistema de Gestión de Inventario. Este documento está destinado a proporcionar una comprensión detallada del diseño, implementación y funcionamiento del sistema desarrollado en Fortran. A lo largo de este manual, se describirán en profundidad los componentes técnicos, las estructuras de datos utilizadas, y los algoritmos implementados, proporcionando así una visión integral para los desarrolladores y técnicos que deseen comprender, mantener o expandir el sistema

## **OBJETIVOS**

### **1. GENERAL**

- 1.1. Orientar a programadores interesados en el sistema de gestión de inventario hecho a base de fortran.

### **2. ESPECÍFICOS**

- 2.1. Explicar de manera detallada cada uno de los archivos principales del programa.
- 2.2. Explicar la lógica detrás de cada uno de las subrutinas creadas y sus funciones.
- 2.3. Permitir el fácil acceso a las funcionalidades del programa.

## ALCANCES DEL SISTEMA

El Sistema de Gestión de Inventario está diseñado para ofrecer una solución robusta para la administración de equipos en inventarios. Las funcionalidades principales del sistema incluyen:

- **Carga de Inventario Inicial:** Importación de datos de inventario desde un archivo, que incluye la lectura y el análisis de datos de equipos con atributos como nombre, cantidad, precio unitario y ubicación.
- **Procesamiento de Movimientos:** Interpretación y ejecución de instrucciones para la actualización del inventario, permitiendo la adición y eliminación de stock.
- **Generación de Informes:** Creación de informes detallados del inventario que muestran la cantidad, precio unitario y valor total de los equipos.

## ESPECIFICACIÓN TÉCNICA

### 1. REQUISITOS DE HARDWARE

- 2 Gb de memoria Ram
- 500MB de almacenamiento interno
- Computadora
- Tarjeta de video integrada

### 2. REQUISITOS DE SOFTWARE

- Visual estudio code
- Windows 11
- Paquete de Fortran

## ARQUITECTURA DE SOFTWARE

El sistema está estructurado en varios módulos interrelacionados que facilitan la separación de responsabilidades y el mantenimiento del código. La arquitectura del sistema incluye:

- **Módulo InventarioModule:** Define el tipo de datos Equipo, que encapsula la información sobre cada equipo en el inventario.
- **Módulo GestorInventario:** Contiene subrutinas para la carga del inventario, procesamiento de movimientos, y generación de informes. También incluye funciones auxiliares para la gestión y manipulación de datos.
- **Programa Principal:** Implementa la lógica de control del sistema, permitiendo a los usuarios interactuar con el sistema a través de un menú de opciones.

# LÓGICA DEL PROGRAMA

- **MODULO DE GESTORINVENTARIO:**

## 1. Subrutina cargarInventario

### Propósito:

Esta subrutina se encarga de cargar el inventario inicial desde un archivo de texto. Lee las líneas del archivo y llama a la subrutina procesarInstruccion para procesar cada línea.

### Lógica:

- **Abrir Archivo:** Se abre el archivo especificado por file\_name para lectura.
- **Leer Línea por Línea:** Se leen las líneas del archivo hasta que se alcance el final. Cada línea se procesa si no está vacía.
- **Procesar Línea:** Se llama a procesarInstruccion con la línea leída y el delimitador ';' para dividir y procesar los datos.
- **Cerrar Archivo:** Una vez que se ha leído todo el archivo, se cierra.

### Errores Manejo:

- Se maneja el error si el archivo no puede abrirse, terminando la ejecución con un mensaje de error.

## 2. Subrutina procesarMovimientos

### Propósito:

Esta subrutina procesa un archivo de movimientos para actualizar el inventario existente con las instrucciones de modificaciones.

### Lógica:

- **Abrir Archivo:** Se abre el archivo de movimientos especificado por file\_name.
- **Leer Línea por Línea:** Similar a cargarInventario, se leen y procesan las líneas hasta el final del archivo.
- **Procesar Línea:** Cada línea se pasa a procesarInstruccion para aplicar las modificaciones al inventario.
- **Cerrar Archivo:** El archivo se cierra al final del proceso.

### Errores Manejo:

- Similar a cargarInventario, se maneja el error si el archivo no se puede abrir.

## 3. Subrutina extendedArray

**Propósito:**

Esta subrutina redimensiona un arreglo de Equipo para permitir el crecimiento del inventario.

**Lógica:**

- **Determinar Tamaño Nuevo:** Calcula el nuevo tamaño del arreglo añadiendo un incremento fijo (10 en este caso).
- **Redimensionar Arreglo:** Crea un nuevo arreglo temporal, copia los datos del arreglo original y lo reemplaza con el nuevo arreglo.

**Errores Manejo:**

- Manejo de errores en caso de problemas de asignación de memoria no está explicitado pero se recomienda revisar el estado de allocate.

**4. Subrutina extendedArrayCaracteres****Propósito:**

Redimensiona un arreglo de caracteres. Funciona de manera similar a extendedArray pero para arreglos de caracteres.

**Lógica:**

- **Determinar Tamaño Nuevo:** Similar a extendedArray, calcula el nuevo tamaño y redimensiona el arreglo de caracteres.

**Errores Manejo:**

- Manejo de errores en la asignación de memoria se recomienda verificar.

**5. Subrutina procesarInstruccion****Propósito:**

Procesa una línea de instrucción y actualiza el inventario de acuerdo con la operación especificada.

**Lógica:**

- **Separar Datos:** Utiliza separarDatos para dividir la línea en operación y datos.
- **Verificar Operación:** Dependiendo del tipo de operación (crear\_equipo, agregar\_stock, eliminar\_equipo), se realizan diferentes acciones:
  - **Crear Equipo:** Extrae datos del equipo y lo agrega al inventario.
  - **Agregar Stock:** Busca el equipo y aumenta su cantidad.
  - **Eliminar Equipo:** Busca el equipo y disminuye su cantidad.
- **Error Manejo:** Muestra mensajes de error si la operación no es válida o si el equipo no se encuentra en el inventario.

**Errores Manejo:**

- Manejo de errores de lectura y de operación no válida.

## 6. Subrutina separarDatos

### Propósito:

Separa una línea de texto en operación y datos utilizando un delimitador.

### Lógica:

- **Separar Operación:** Extrae la operación que precede el primer delimitador.
- **Contar Delimitadores:** Determina el número de partes en los datos.
- **Dividir Datos:** Separa los datos según el delimitador y los almacena en un arreglo.

### Errores Manejo:

- Manejo de errores no está explicitado pero el proceso de separación asume datos bien formados.

## 7. Subrutina reshapeDatos

### Propósito:

Ajusta el tamaño del arreglo de datos para que coincida con el número de palabras especificado.

### Lógica:

- **Redimensionar Arreglo:** Crea un nuevo arreglo con el tamaño correcto y copia los datos necesarios.

### Errores Manejo:

- Asegúrese de que el tamaño del nuevo arreglo sea adecuado y maneje cualquier error potencial en la asignación de memoria.

## 8. Subrutina obtenerCantidadPrecio

### Propósito:

Extrae la cantidad y el precio unitario de un arreglo de datos.

### Lógica:

- **Leer Cantidad:** Lee la cantidad del segundo dato.
- **Leer Precio Unitario:** Lee el precio unitario si está disponible en el tercer dato.

### Errores Manejo:

- Manejo de errores en lectura y validación de datos.

## 9. Subrutina crearInforme



**Propósito:**

Genera un informe detallado del inventario y lo guarda en un archivo.

**Lógica:**

- **Abrir Archivo:** Se abre un archivo para escritura del informe.
- **Escribir Encabezado:** Escribe los encabezados y formato del informe.
- **Escribir Datos del Inventario:** Recorre el inventario y escribe cada entrada en el archivo.

**Errores Manejo:**

- Manejo de errores al abrir el archivo para escritura no está explicitado pero se recomienda revisar la asignación de recursos.

**10. Subrutina separar****Propósito:**

Separa una línea de texto en componentes de un equipo.

**Lógica:**

- **Separar Línea:** Utiliza un delimitador para dividir la línea en tokens.
- **Asignar Valores:** Asigna los tokens a las propiedades del equipo temporal.

**Errores Manejo:**

- Manejo de errores en el caso de tokens mal formados o insuficientes.

- **MAIN.f90:**

En este archivo solo se encuentra la lógica principal y llama las funciones desde el menú principal, luego de seleccionar la opción deja que el modulo GestorInventario se encargue de las funciones y se regresa al menú cada vez que termina su función.

- **INVENTARIOMODULE.f90:**

Este archivo almacena el objeto principal de nuestro programa el objeto “Equipo” y sus atributos.