

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
LENGUAJES FORMALES Y DE PROGRAMACION
CATEDRÁTICO: ING. ZULMA AGUIRRE
TUTOR ACADÉMICO: JONATAN LEONEL GARCIA ARANA



Brandon Antonio Marroquin Pérez

CARNÉ: 202300813

SECCIÓN: B-

GUATEMALA, 22 DE AGOSTO DEL 2,024

ÍNDICE

| | |
|---|---|
| INTRODUCCIÓN | 2 |
| OBJETIVOS | 2 |
| 1. GENERAL | 2 |
| 2. ESPECÍFICOS | 2 |
| ALCANCES DEL SISTEMA..... | 3 |
| ESPECIFICACIÓN TÉCNICA..... | 4 |
| • REQUISITOS DE HARDWARE | 4 |
| • REQUISITOS DE SOFTWARE..... | 4 |
| DESCRIPCIÓN DE LA SOLUCIÓN | 5 |
| LÓGICA DEL PROGRAMA..... | 6 |
| ➤ Variables Globales y Su Función..... | 6 |
| ➤ Variables Locales y Su Función..... | 7 |
| ➤ Tipos de Datos y Su Función..... | 8 |
| ➤ Subprogramas (Subrutinas) y Su Función:..... | 8 |
| ➤ PROGRAMpractical..... | 9 |

INTRODUCCIÓN

Este Manual Técnico describe el desarrollo de un Sistema de Gestión de Inventarios de Equipos de Oficina utilizando el lenguaje Fortran. El objetivo principal de este sistema es ofrecer una solución eficiente para la administración de inventarios, permitiendo registrar y gestionar movimientos de equipos mediante archivos de texto. A través de la manipulación de estructuras de datos y el uso de algoritmos de búsqueda y ordenamiento, el sistema garantiza un manejo preciso y actualizado del inventario, facilitando así la gestión de recursos en oficinas y otros entornos organizacionales.

OBJETIVOS

1. GENERAL

- 1.1. Desarrollar un Sistema de Gestión de Inventarios en Fortran que permita la administración eficaz de equipos de oficina, mediante el manejo de archivos de texto y la implementación de estructuras de datos que aseguren un control preciso y ordenado del inventario.

2. ESPECÍFICOS

- 2.1. Implementar algoritmos de búsqueda y ordenamiento que optimicen la localización y organización de los equipos en el inventario.
- 2.2. Desarrollar rutinas de manipulación de archivos que permitan registrar, actualizar y eliminar entradas en el inventario, asegurando la integridad de los datos almacenados.

ALCANCES DEL SISTEMA

El Sistema de Gestión de Inventarios de Equipos de Oficina desarrollado en Fortran se enfoca en la administración precisa y eficiente de recursos, abarcando el registro, actualización y eliminación de equipos en un inventario centralizado. Este sistema permite la lectura y escritura de archivos de texto para manejar movimientos de inventario, como adiciones y eliminaciones de stock, y asegura la integridad de los datos mediante algoritmos de búsqueda y ordenamiento. Además, incluye una interfaz de usuario en consola que facilita la interacción con el sistema, orientada a usuarios encargados de la administración de recursos en entornos organizacionales. El manual técnico proporciona instrucciones detalladas para la instalación, configuración y operación del sistema, garantizando su correcta implementación y uso eficiente.

ESPECIFICACIÓN TÉCNICA

● REQUISITOS DE HARDWARE

- **Sistema operativo:**
 - Windows 10 o superior
 - macOS 10.14 o superior
 - Ubuntu 18.04 o superior
- **Procesador:**
 - Intel Core i5 o superior
 - AMD Ryzen 5 o superior
- **Memoria RAM:**
 - 12 GB o superior
- **Almacenamiento:**
 - 5 GB de espacio libre en disco
- **Tarjeta gráfica:**
 - Intel HD Graphics 620 o superior
 - AMD Radeon R5 o superior
- **Pantalla:**
 - Resolución de 1280 x 720 o superior
- **Software:**
 - Visual Studio Code
 - Fortran
 - Git (opcional para control de versiones)

● REQUISITOS DE SOFTWARE

- **Software esencial:**
 - Visual Studio Code: Es el entorno de desarrollo integrado (IDE) utilizado para escribir, depurar y ejecutar el código Fortran.
 - Última versión de Fortran: Es el compilador necesario para compilar y ejecutar el código Fortran.

- Git (opcional para control de versiones): Es un sistema de control de versiones que permite realizar un seguimiento de los cambios en el código fuente y colaborar con otros desarrolladores.

DESCRIPCIÓN DE LA SOLUCIÓN

Primero, se desarrolló el módulo de gestión de inventario, que permite la carga inicial del inventario de equipos de oficina desde un archivo .inv. Este módulo es fundamental ya que administra las operaciones de carga y mantiene un registro detallado de los equipos. A continuación, se implementó el sistema de movimientos de inventario, donde se manejan instrucciones para agregar y eliminar stock de equipos a través de archivos .mov. Se crearon subrutinas para interpretar y procesar estos archivos, ajustando las cantidades y generando reportes que reflejan el estado actual del inventario. Finalmente, se implementó la generación de informes, que consolida la información del inventario en un archivo .txt, mostrando detalles como el nombre del equipo, cantidad disponible, precio unitario, valor total y ubicación.

LÓGICA DEL PROGRAMA

LFP_S2_2024_Practica1_202300813

Variables Globales y Su Función:

MAX_INVENTARIO:

Tipo: INTEGER

Función: Es una constante que define el tamaño máximo permitido para el inventario. Sirve para limitar la cantidad de elementos que se pueden almacenar en el inventario.

inventarios:

Tipo: TYPE(Inventario), DIMENSION(:), ALLOCATABLE

Función: Es un array dinámico que contiene los registros de los inventarios. Almacena todos los equipos, sus cantidades, precios y ubicaciones.

num_inventario:

Tipo: INTEGER

Función: Se utiliza para almacenar el número máximo de inventarios que el programa puede manejar, que en este caso está dado por MAX_INVENTARIO.

opcion:

Tipo: INTEGER

Función: Almacena la opción seleccionada por el usuario en el menú principal del programa.

contador:

Tipo: INTEGER

Función: Lleva la cuenta del número actual de inventarios que se han registrado. Es utilizado para indexar los elementos en el array inventarios.

filename:

Tipo: CHARACTER(LEN=100)

Función: Almacena el nombre del archivo desde el cual se leerá el inventario inicial o las instrucciones de movimiento.

Variables Locales y Su Función:

line:

Tipo: CHARACTER(LEN=200)

Función: En los subprogramas analizador, extraer_datos_mov, parse_line, y generar_informe, line es utilizado para almacenar temporalmente la línea leída desde un archivo.

comando y datos:

Tipo: CHARACTER(LEN=100)

Función: En el subprograma analizador, comando almacena la acción a realizar (por ejemplo, agregar o eliminar stock), y datos almacena los parámetros asociados a esa acción.

nombre, ubicacion, cantidad, precio_unitario:

Tipo: CHARACTER(LEN=100) para nombre y ubicacion, INTEGER para cantidad, REAL para precio_unitario

Función: Estas variables almacenan los atributos de un inventario o las modificaciones que se realizarán sobre un inventario existente.

ios:

Tipo: INTEGER

Función: Es una variable de estado de entrada/salida utilizada para verificar si las operaciones de lectura y escritura se ejecutaron correctamente.

temp_line, field, temp_int, temp_real:

Tipo: CHARACTER(LEN=200) para temp_line, CHARACTER(LEN=300) para field, INTEGER para temp_int, REAL para temp_real

Función: En parse_line, estas variables se usan para procesar y almacenar temporalmente las partes de una línea que se están analizando.

encontrado:

Tipo: LOGICAL

Función: En los subprogramas agregar_stock y eliminar_equipo, indica si el equipo buscado fue encontrado en el inventario.

Tipos de Datos y Su Función:

❖ TYPE :: Inventario:

Función: Define la estructura de un inventario, que incluye el nombre, ubicación, cantidad y precio unitario del equipo. Este tipo de dato es utilizado para crear el array inventarios que almacena todos los equipos en el inventario.

Subprogramas (Subrutinas) y Su Función:

❖ analizador(archivo, es_mov, contador):

Función: Lee un archivo de inventario o de movimientos y lo procesa. Si es un archivo de inventario (es_mov es .FALSE.), lo carga en el array inventarios. Si es un archivo de movimientos (es_mov es .TRUE.), procesa las instrucciones para agregar o eliminar stock.

❖ parse_line(line, inventario1):

Función: Extrae los valores de una línea de texto que contiene los datos de un inventario (nombre, cantidad, precio_unitario, ubicacion) y los almacena en una instancia del tipo Inventario.

❖ extraer_datos_mov(datos, nombre, cantidad, ubicacion):

Función: Separa una línea de datos de movimiento en sus componentes (nombre, cantidad, ubicacion) para ser procesados por las subrutinas de agregar_stock o eliminar_equipo.

❖ generar_informe(inventarios, num_inventario):

Función: Crea un informe en un archivo de texto con el estado actual del inventario, mostrando el nombre del equipo, la cantidad disponible, el precio unitario, el valor total y la ubicación.

❖ **agregar_stock(inventarios, contador, nombre, cantidad, ubicacion):**

Función: Aumenta la cantidad de un equipo específico en el inventario, si este ya existe en la ubicación dada. Si no existe, se muestra un error.

❖ **eliminar_equipo(inventarios, contador, nombre, cantidad, ubicacion):**

Función: Reduce la cantidad de un equipo específico en el inventario, si este existe en la ubicación dada y hay suficiente stock para realizar la eliminación.

PROGRAM practica1: Es el punto de entrada del programa. Aquí es donde se arranca la ejecución principal del código. Se encarga de gestionar el flujo general del programa, incluyendo la presentación del menú de opciones, la inicialización de variables, y la llamada a subrutinas como analizador, generar_informe, agregar_stock, y eliminar_equipo.