LENGUAJES FORMALES Y DE PROGRAMACION

MANUAL TÉCNICO

BRAYAN ALEXANDER GUZMAN MARGOS 202105658

INVENTARIO

Contenido

I. Introducción

Este programa, desarrollado en el lenguaje Fortran, está diseñado como parte del curso de Lenguajes Formales y de Programación. Su propósito es gestionar un inventario de equipos de oficina, permitiendo registrar y controlar los movimientos de los mismos a través del uso de archivos de texto.

El sistema implementado facilita el seguimiento de los equipos disponibles, los que han sido asignados, y aquellos que han sido devueltos o retirados. Además, ofrece una interfaz sencilla para la actualización y consulta del inventario, asegurando que toda la información relevante se mantenga organizada y accesible.

II. Objetivos

Desarrollar un programa en lenguaje Fortran que permita gestionar de manera eficiente un inventario de equipos de oficina, registrando los movimientos de ingreso, asignación y retiro de dichos equipos mediante el uso de archivos de texto, con el fin de proporcionar a los usuarios una herramienta fiable para el control y seguimiento de los recursos disponibles.

III. Alcances del Sistema

El sistema desarrollado en Fortran tiene como alcance principal la gestión integral de un inventario de equipos de oficina mediante el uso de archivos de texto. Permitirá registrar y actualizar los movimientos de equipos, como ingresos, asignaciones y retiros, manteniendo un registro detallado y preciso de todos los recursos. Los usuarios podrán consultar el estado actual del inventario y el historial de movimientos a través de una interfaz simple e intuitiva, sin necesidad de conocimientos técnicos avanzados. Además, el sistema facilitará la generación de reportes básicos que ofrecerán información clave para la toma de decisiones y la administración eficiente de los equipos.

IV. Especificación Técnica

1. Requisitos de Hardware

- Computadora de escritorio o portátil.
- Mínimo 4GB de Memoria RAM.
- 128 GB de almacenamiento de Disco Duro o un SSD para un mejor rendimiento.
- Procesador Intel Core i5 o Ryzen 5.

2. Requisitos de Software

- Sistema Operativo Windows 10 o superior. Tener instalado Visual en la versión 17.7.5
- Tener instalada Fortran.
- Editor de Texto como Visual Studio Code.

V. DESCRIPCIÓN DE LA SOLUCIÓN

El código en Fortran proporcionado desarrolla un sistema para gestionar un inventario de equipos de oficina, con la capacidad de registrar y controlar los movimientos de los equipos mediante archivos de texto. El programa está estructurado en módulos que definen las características y operaciones de los equipos, como la adición y eliminación de stock. Utiliza dos módulos principales: `InventarioModule`, que define el tipo de datos `Equipo` y sus procedimientos asociados, y `global_vars`, que maneja el inventario global. El programa principal ofrece un menú interactivo para cargar inventarios iniciales, procesar movimientos de equipos, y generar informes. Los archivos de inventario y movimientos se leen para actualizar los datos del sistema y mantener un seguimiento preciso de los equipos.

VI. Logica del Programa

PROGRAMA

- `program main` Esta línea define el inicio de un módulo llamado InventarioModule

SUBRUTINAS

1) La subrutina crearEquipo toma la información de un nuevo equipo (nombre, ubicación, cantidad, y precio unitario), crea un objeto Equipo, lo inicializa con esos valores, y luego lo almacena en el inventario global. Finalmente, actualiza el contador n para reflejar la nueva cantidad de equipos en el inventario. Esto permite que el sistema gestione dinámicamente la incorporación de nuevos equipos.

```
subroutine crearEquipo(nombre, ubicacion, cantidad, precio_unitario)
    use InventarioModule
    use global_vars
    character(len=256), intent(in) :: nombre, ubicacion
    integer, intent(in) :: cantidad
    real, intent(in) :: precio_unitario

    type(Equipo) :: nuevoEquipo
    call nuevoEquipo%inicializar(nombre, ubicacion, cantidad, precio_unitario)
    inventario(n) = nuevoEquipo
    n = n + 1
end subroutine crearEquipo
```

2) La subrutina cargarInstruccionesMovimientos es responsable de leer un archivo de texto que contiene instrucciones para mover el inventario, como agregar o eliminar stock de equipos. Primero, abre el archivo indicado por el usuario y, línea por línea, interpreta las instrucciones. Cada línea del archivo especifica un comando (por ejemplo, agregar_stock o eliminar_equipo), seguido del nombre del equipo, la cantidad, y la ubicación. Según el comando, la subrutina llama a otras subrutinas (aumentarStock o borraStock) para ejecutar la acción correspondiente sobre el equipo en el inventario. Si el archivo no puede abrirse, se muestra un mensaje de error. Al finalizar la lectura y ejecución de las instrucciones, cierra el archivo y notifica al usuario que las instrucciones se han cargado correctamente.

```
subroutine cargarInstruccionesMovimientos()
    use InventarioModule
    use global_vars
    integer :: iunit, ios, pos, cantidad_int
    character(len=256) :: linea, comando, nombre, ubicacion, cantidad
    character(len=100) :: nombre_archivo_acciones

    iunit = 11

    print*, "Ingresa el nombre del archivo que quieres abrir con .mov"
    read(*,*) nombre_archivo_acciones

    open(unit=iunit, file=trim(nombre_archivo_acciones), status="old", action="read", iostat=ios)
    if (ios /= 0) then
        print *, "Error no se puede abrir el archivo de movimientos :("
        return
    endif
```

3) La subrutina aumentarStock en Fortran se encarga de incrementar la cantidad de un equipo específico en el inventario. Primero, recibe como parámetros el nombre, la ubicación del equipo y la cantidad a agregar. Luego, busca en el inventario global el equipo que coincida con el nombre y la ubicación proporcionados. Si encuentra el equipo, llama al método agregarStock del objeto Equipo para aumentar la cantidad. Si no encuentra el equipo en la ubicación especificada, muestra un mensaje de error indicando que el equipo no existe en esa ubicación. Esta subrutina permite actualizar las existencias de un equipo en el inventario de manera controlada.

```
subroutine aumentarStock(nombre, ubicacion, cantidad)
    use InventarioModule
    use global_vars
    character(len=256), intent(in) :: nombre, ubicacion
    integer, intent(in) :: cantidad

    integer :: i
    logical :: encontrado = .false.

    do i = 1, n-1
        if (inventario(i)%nombre == nombre .and. inventario(i)%ubicacion == ubicacion) then
            call inventario(i)%agregarStock(cantidad)
            encontrado = .true.
        endif
    end do

    if (.not. encontrado) then
        print *, "Error: El equipo no se encuentra en la ubicacion indicada :("
    endif
end subroutine aumentarStock
```

- *MODULOS*

- El módulo InventarioModule define la estructura y operaciones de un equipo de oficina en el inventario. Contiene la definición del tipo Equipo, que encapsula propiedades como nombre, ubicación, cantidad y precio unitario. Además, proporciona métodos (subrutinas) para inicializar un equipo, agregar y quitar stock, permitiendo manipular cada equipo como un objeto con estas operaciones asociadas.

```
module InventarioModule
    implicit none
```

- El módulo global_vars, por otro lado, gestiona las variables globales que son utilizadas en el programa. Específicamente, define un arreglo inventario de hasta 100 equipos y un contador n que lleva el registro de cuántos equipos están actualmente almacenados. Este módulo

centraliza el almacenamiento de los datos del inventario, permitiendo que sean accesibles desde diferentes partes del programa.

```
module global_vars
    use InventarioModule
```