UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS
LENGUAJES FORMALES DE PROGRAMACION
SEGUNDO SEMESTRE 2024

MANUAL TECNICO

PRACTICA 1

Nombre: Alexander Samuel Us Upún
Carné 202300824
Fecha: 22/08/2024

Programa Principal: main

Este es el programa principal que utiliza el módulo inventario para gestionar el inventario de un sistema. Ofrece un menú interactivo con varias opciones para que el usuario cargue el inventario inicial, procese movimientos de stock, y genere un informe del inventario.

Variables principales:

- **integer :: option**: Almacena la opción seleccionada por el usuario en el menú.
- **character(len=100) :: archivo**: Almacena el nombre del archivo que se ingresa para cargar el inventario o los movimientos.
- **type(inicial), allocatable :: inventarioInicial(:)**: Arreglo dinámico que guarda los datos del inventario inicial cargado desde un archivo.
- **type(movimiento), allocatable :: movimientos(:)**: Arreglo dinámico que guarda los movimientos cargados desde un archivo.

Ciclo Principal do while (.true.):

El programa entra en un bucle infinito que presenta un menú con varias opciones. El ciclo continúa hasta que el usuario elige salir del programa.

1. **Menú de Opciones**:
    - El programa imprime un menú interactivo con las siguientes opciones:
        1. **Cargar inventario inicial**: Permite al usuario cargar un inventario desde un archivo.
        2. **Cargar instrucciones de movimientos**: Permite al usuario cargar y procesar movimientos de stock desde un archivo.
        3. **Crear informe de inventario**: Genera un archivo de texto con un informe del inventario actual.
        4. **Salir**: Termina el programa.
2. **Entrada de Usuario**:
    - El programa solicita al usuario que ingrese una opción y almacena la selección en la variable option.
    - Utiliza una estructura select case para ejecutar la acción correspondiente según la opción seleccionada.

Opciones del Menú:

1. **Caso 1: Cargar Inventario Inicial**
    - El programa solicita el nombre del archivo que contiene los datos del inventario inicial.

- Llama a la subrutina cargar_y_procesar_inventario del módulo inventario para cargar y procesar los datos desde el archivo especificado.
- Una vez completado, se imprime un mensaje confirmando que el inventario se ha cargado correctamente.

2. **Caso 2: Cargar Instrucciones de Movimientos**
   - El programa solicita el nombre del archivo que contiene las instrucciones de movimientos.
   - Llama a la subrutina cargar_movimientos para procesar los movimientos y actualizar el inventario inicial en consecuencia.
   - Durante este proceso, se imprimen mensajes que indican el estado de cada movimiento procesado.
3. **Caso 3: Crear Informe de Inventario**
   - Llama a la subrutina imprimir_inventario para generar un informe del inventario actual y guardarlo en un archivo llamado reporte.txt.
   - El archivo reporte.txt contiene los detalles del inventario, incluyendo el nombre del equipo, la cantidad, el precio unitario, el valor total, y la ubicación.
4. **Caso 4: Salir**
   - El programa imprime un mensaje de salida y termina el ciclo, finalizando la ejecución del programa.
5. **Caso default: Opción no válida**
   - Si el usuario ingresa un número que no corresponde a ninguna opción válida (1-4), el programa imprime un mensaje de error y vuelve a mostrar el menú.

Módulo inventario

El módulo inventario define dos tipos de datos (inicial y movimiento) y varias subrutinas para manejar inventarios y movimientos de stock.

Tipos de datos:

1. **Tipo inicial:**
   - **Descripción**: Representa un equipo en el inventario inicial.
   - **Componentes**:
     - character(len=100) :: nombre: El nombre del equipo.
     - integer :: cantidad: La cantidad de unidades disponibles.

        - real :: precio_unitario: El precio por unidad del equipo.
        - real :: precio_total: El precio total (cantidad * precio_unitario).
        - character(len=50) :: ubicacion: La ubicación del equipo en el inventario.
2. **Tipo movimiento:**
    - **Descripción**: Representa un movimiento de stock (agregar o eliminar).
    - **Componentes**:
        - character(len=100) :: nombre: El nombre del equipo involucrado en el movimiento.
        - integer :: cantidad: La cantidad de unidades que se mueven (agregadas o eliminadas).
        - character(len=50) :: ubicacion: La ubicación del equipo donde se realiza el movimiento.

Subrutinas:

1. **Subrutina cargar_y_procesar_inventario:**
    - **Descripción**: Carga los datos del inventario inicial desde un archivo y los procesa para llenar una lista de equipos.
    - **Parámetros**:
        - character(len=*) :: nombre_archivo: Nombre del archivo que contiene el inventario inicial.
        - type(inicial), allocatable :: inventarioInicial(:): Arreglo dinámico para almacenar el inventario cargado.
    - **Funcionamiento**:
        - Cuenta el número de equipos en el archivo y reserva memoria para almacenarlos.
        - Lee cada línea del archivo, descompone los datos en nombre, cantidad, precio unitario y ubicación, y luego los almacena en inventarioInicial.
        - Calcula el precio total por equipo y lo almacena.
        - Finalmente, imprime el inventario en la consola.
2. **Subrutina cargar_movimientos:**
    - **Descripción**: Carga y procesa movimientos de stock desde un archivo, actualizando el inventario inicial según sea necesario.
    - **Parámetros**:

        - character(len=*) :: nombre_archivo: Nombre del archivo que contiene los movimientos.
        - type(movimiento), allocatable :: movimientos(:): Arreglo dinámico para almacenar los movimientos cargados.
        - type(inicial), intent(inout) :: inventarioInicial(:): Arreglo del inventario inicial que se actualizará con los movimientos.
    - **Funcionamiento**:
        - Cuenta el número de movimientos en el archivo y reserva memoria para almacenarlos.
        - Lee cada línea, descompone los datos en nombre, cantidad y ubicación, y busca el equipo correspondiente en el inventario inicial.
        - Si se encuentra el equipo:
            - **Si la instrucción es agregar_stock**: Suma la cantidad especificada al inventario existente.
            - **Si la instrucción es eliminar_equipo**: Resta la cantidad especificada, siempre que la cantidad a eliminar no exceda la cantidad existente.
        - Imprime mensajes para indicar los cambios realizados o errores si no se puede procesar un movimiento.
3. **Subrutina imprimir_inventario_consola:**
    - **Descripción**: Imprime el inventario inicial en la consola en formato de tabla.
    - **Parámetros**:
        - type(inicial), intent(in) :: inventarioInicial(:): Arreglo del inventario inicial a imprimir.
    - **Funcionamiento**:
        - Imprime una cabecera para la tabla.
        - Recorre el inventario inicial y muestra cada equipo, cantidad, precio unitario, precio total y ubicación.
4. **Subrutina imprimir_inventario:**
    - **Descripción**: Guarda el inventario inicial en un archivo de texto.
    - **Parámetros**:
        - type(inicial), intent(in) :: inventarioInicial(:): Arreglo del inventario inicial a guardar.

        - character(len=*) :: nombre_archivo: Nombre del archivo donde se guardará el inventario.
    - **Funcionamiento**:
        - Abre un archivo y escribe una cabecera.
        - Recorre el inventario inicial y guarda cada equipo, cantidad, precio unitario, precio total y ubicación en el archivo.