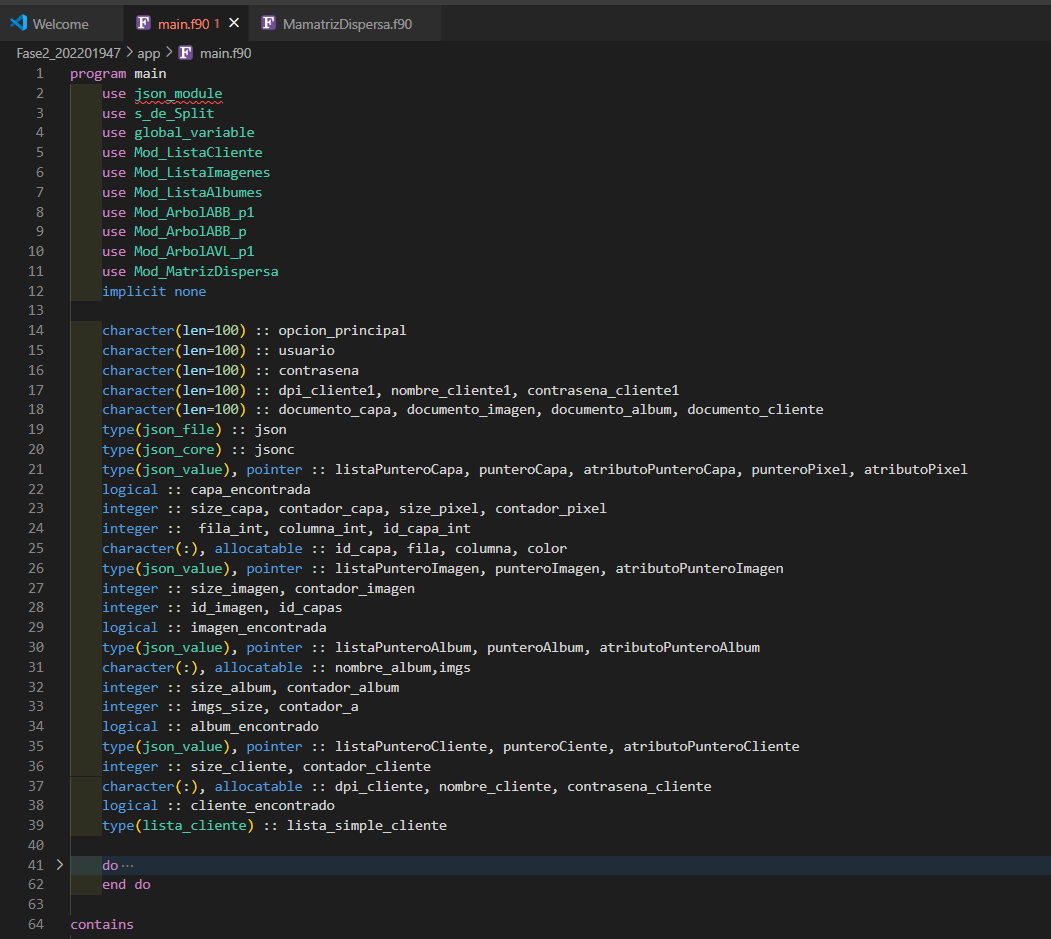
MENU PRINCIPAL

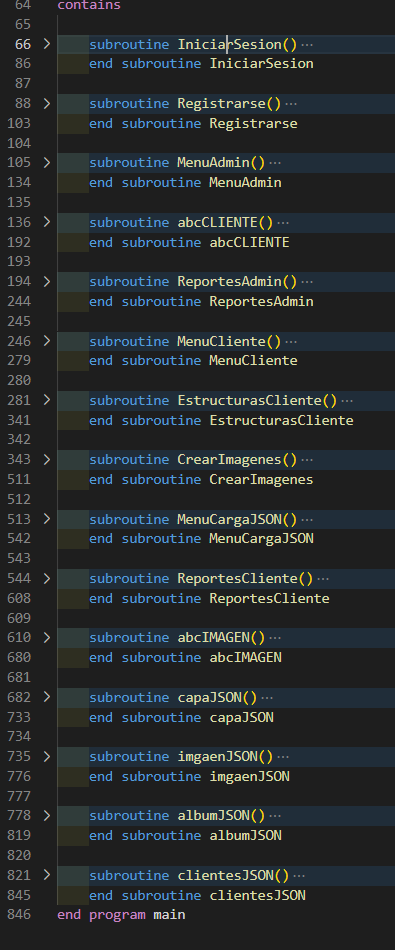
**Uso de las importaciones de las estructuras:**

En esta sección, se detalla el uso adecuado de las importaciones de las estructuras necesarias para el funcionamiento del proyecto. Las importaciones proporcionan acceso a las bibliotecas y módulos externos requeridos para implementar funcionalidades específicas. A continuación, se describen las principales estructuras que deben importarse y su propósito en el proyecto. Además, se explican las buenas prácticas para organizar las importaciones y garantizar la modularidad y legibilidad del código.

**Declaración de variables para el menú y para la lectura de los JSON:**

Esta sección aborda la declaración apropiada de variables necesarias para la implementación del menú y la lectura de archivos JSON en el proyecto. Las variables desempeñan un papel fundamental en el almacenamiento temporal de datos y en la configuración de opciones disponibles para el usuario. Se detallan las convenciones de nomenclatura y los tipos de datos recomendados para cada variable, así como su ámbito de aplicación dentro del código.

Subrutinas del menú principal:

Esta sección detalla las subrutinas que componen el menú principal del proyecto en Fortran. El menú principal incluye opciones para iniciar sesión como cliente o administrador, así como las funcionalidades específicas asociadas con cada rol. Se describen las subrutinas relevantes para el proceso de inicio de sesión, la navegación del menú y la ejecución de las acciones correspondientes a cada rol.

**Inicio de sesión:**

Se describe la subrutina encargada de gestionar el proceso de inicio de sesión, donde los usuarios ingresan sus credenciales para acceder al sistema como cliente o administrador. Se explican los pasos necesarios para validar las credenciales proporcionadas por el usuario y dirigirlos al menú correspondiente según su rol.

**Menú del cliente:**

Se detallan las subrutinas que forman parte del menú destinado a los clientes una vez que han iniciado sesión. Este menú proporciona opciones específicas para que los clientes gestionen sus álbumes, carguen capas de imágenes y realicen otras acciones relacionadas con su cuenta. Se describen las funcionalidades disponibles y cómo los clientes pueden interactuar con el sistema para llevar a cabo sus tareas.

**Menú del administrador:**

Se describen las subrutinas que componen el menú diseñado para los administradores después de iniciar sesión. Este menú ofrece opciones para que los administradores gestionen clientes, carguen información en la base de datos y realicen otras tareas administrativas relacionadas con el sistema. Se explican las restricciones y permisos específicos asociados con el rol de administrador, como la capacidad exclusiva de cargar clientes en la base de datos.

**Funcionalidades específicas:**

Se detallan las funcionalidades exclusivas de cada rol, como la capacidad de los administradores para cargar clientes en el sistema y la capacidad de los clientes para cargar capas de imágenes y álbumes. Se explican las restricciones y validaciones asociadas con estas funcionalidades para garantizar la integridad y seguridad de los datos en el sistema.

Matriz Dispersa

Una matriz dispersa en Fortran es una estructura de datos eficiente que almacena solo los elementos no nulos junto con sus ubicaciones (fila, columna). Enumerando sus propiedades:

1. **Eficiencia de almacenamiento:** Almacena solo elementos no nulos, reduciendo el uso de memoria en comparación con matrices densas.

**Subrutina para insertar un elemento:** Esta subrutina permite insertar un nuevo elemento en la matriz dispersa, especificando su valor, fila y columna. Si el elemento ya existe en la matriz, se actualiza su valor.

**Subrutinas para gestionar la cabecera de filas y columnas:** Estas subrutinas ayudan a mantener la estructura de la matriz dispersa, incluyendo la gestión de los punteros que apuntan al inicio de cada fila y columna.

**Subrutina para buscar una fila:** Esta subrutina busca una fila específica en la matriz dispersa y devuelve un puntero al primer elemento de esa fila.

**Función para verificar la existencia de un nodo**: Esta función verifica si un nodo específico existe en la matriz dispersa, dada su fila y columna. Devuelve un valor lógico que indica si el nodo está presente o no.

**Función para graficar la matriz dispersa:** Esta función grafica la matriz dispersa, mostrando sus elementos no nulos y su ubicación en relación con las filas y columnas. Esta función puede utilizar bibliotecas gráficas externas o generar una representación visual simple dentro de la consola.

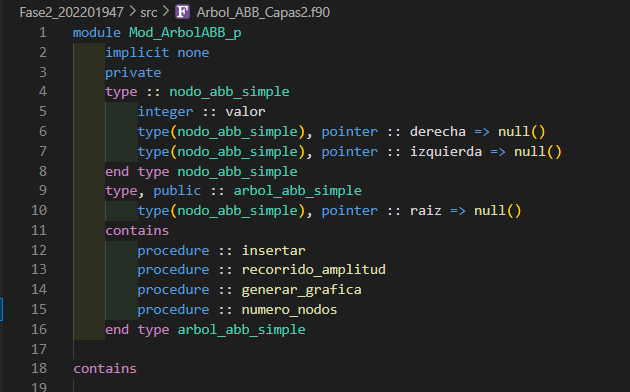
ARBOL ABB

Para incorporar un Árbol Binario de Búsqueda (ABB) en tu proyecto de Fortran, que contiene una matriz dispersa en cada hoja, puedes implementar las siguientes subrutinas y funciones:

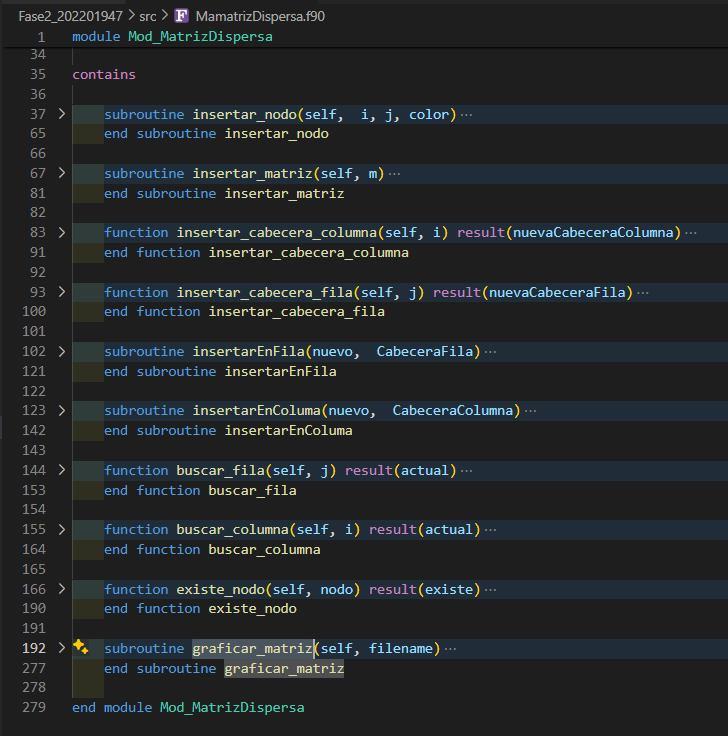
1. **Subrutinas de recorrido:** Recorrido preorden: Esta subrutina recorre el árbol ABB en orden preorden, visitando primero el nodo raíz, luego el subárbol izquierdo y finalmente el subárbol derecho.

Recorrido inorden: Esta subrutina recorre el árbol ABB en orden inorden, visitando primero el subárbol izquierdo, luego el nodo raíz y finalmente el subárbol derecho.

Recorrido postorden: Esta subrutina recorre el árbol ABB en orden postorden, visitando primero el subárbol izquierdo, luego el subárbol derecho y finalmente el nodo raíz.

1. **Subrutina para graficar el árbol:**

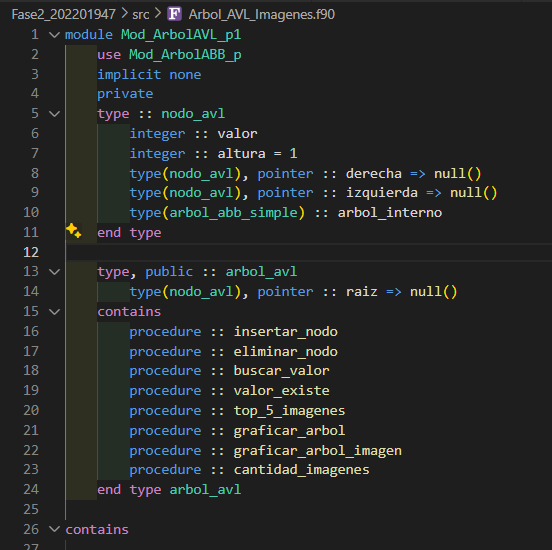
Esta subrutina genera una representación gráfica del árbol ABB, mostrando sus nodos y conexiones entre ellos. Puedes utilizar bibliotecas gráficas externas o generar una representación visual simple dentro de la consola.

1. **Función para calcular la cantidad de capas:**
   * Esta función determina la cantidad total de capas en el árbol ABB, contando desde la raíz hasta la hoja más profunda.
2. **Función para calcular la profundidad del árbol:**

Esta función calcula la profundidad del árbol ABB, es decir, la longitud del camino más largo desde la raíz hasta una hoja.

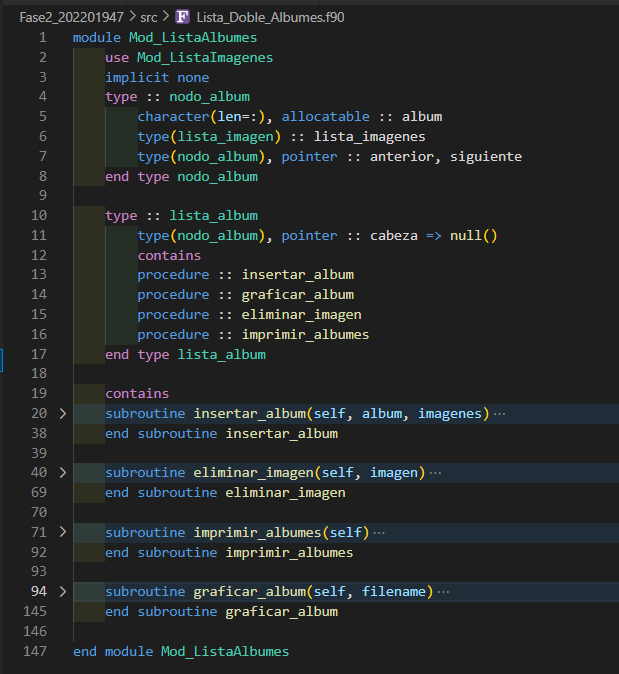
Estas subrutinas y funciones te permitirán trabajar eficazmente con un Árbol Binario de Búsqueda que contiene matrices dispersas en cada hoja. Además, podrás realizar operaciones de recorrido para explorar la estructura del árbol, generar representaciones visuales y obtener información estadística sobre su composición y profundidad.

ARBOL AVL

  
Para integrar un Árbol AVL en tu proyecto Fortran, que almacena imágenes en sus nodos, puedes adaptar las subrutinas y funciones anteriores para que funcionen con este tipo específico de árbol. Aquí te proporciono una descripción de cómo podrían ser estas subrutinas y funciones:

1. **Subrutinas de recorrido:**
   * Recorrido preorden: Recorre el árbol AVL en orden preorden, visitando primero el nodo raíz, luego el subárbol izquierdo y finalmente el subárbol derecho.
   * Recorrido inorden: Recorre el árbol AVL en orden inorden, visitando primero el subárbol izquierdo, luego el nodo raíz y finalmente el subárbol derecho.
   * Recorrido postorden: Recorre el árbol AVL en orden postorden, visitando primero el subárbol izquierdo, luego el subárbol derecho y finalmente el nodo raíz.
2. **Subrutina para graficar el árbol:**
   * Genera una representación visual del árbol AVL, mostrando sus nodos y conexiones entre ellos. Puedes utilizar bibliotecas gráficas externas o generar una representación visual simple dentro de la consola.
3. **Función para calcular la cantidad de imágenes:**
   * Determina la cantidad total de imágenes almacenadas en el árbol AVL, contando los nodos que contienen imágenes.
4. **Función para calcular la profundidad del árbol:**
   * Calcula la profundidad del árbol AVL, es decir, la longitud del camino más largo desde la raíz hasta una hoja.

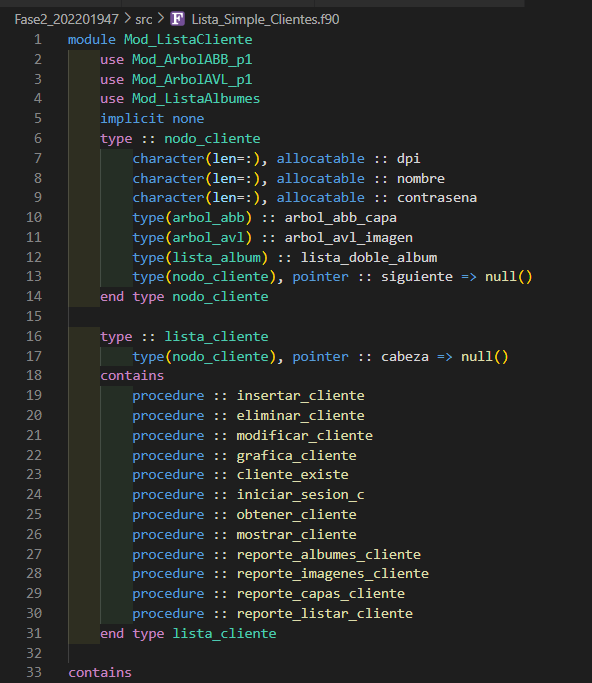
LISTA DOBLEMENTE ENLAZADA

Para manejar álbumes utilizando una lista doblemente enlazada en tu proyecto Fortran, puedes implementar las siguientes subrutinas y funciones:

1. **Subrutina para insertar un álbum:**
   * Esta subrutina permite insertar un nuevo álbum en la lista doblemente enlazada, especificando su nombre u otras características relevantes.
2. **Subrutina para eliminar un álbum:**
   * Elimina un álbum específico de la lista doblemente enlazada, utilizando su nombre u otro identificador único.
3. **Subrutina para graficar la lista de álbumes:**
   * Genera una representación visual de la lista doblemente enlazada de álbumes, mostrando los enlaces entre ellos.
4. **Subrutina para imprimir la lista de álbumes:**
   * Imprime en la consola la información de todos los álbumes presentes en la lista doblemente enlazada, mostrando sus nombres u otras características relevantes.
5. **Subrutina para buscar un álbum:**
   * Busca un álbum específico en la lista doblemente enlazada, utilizando su nombre u otro identificador único, y devuelve un puntero al nodo que lo contiene.

LISTA CLIENTES

Para administrar una lista de clientes en tu proyecto Fortran, puedes utilizar una estructura de datos como una lista enlazada simple. Aquí te muestro cómo podrías implementar las subrutinas necesarias para insertar, modificar, eliminar y graficar clientes en esta lista:

1. **Subrutina para insertar un cliente:**
   * Esta subrutina permite agregar un nuevo cliente a la lista enlazada, proporcionando información como el nombre, ID, dirección de correo electrónico, etc.
2. **Subrutina para modificar un cliente:**
   * Permite actualizar la información de un cliente existente en la lista, identificándolo por su ID u otro identificador único, y proporcionando los nuevos datos.
3. **Subrutina para eliminar un cliente:**
   * Elimina un cliente específico de la lista enlazada, utilizando su ID u otro identificador único.
4. **Subrutina para graficar la lista de clientes:**
   * Genera una representación visual de la lista de clientes, mostrando los enlaces entre ellos y posiblemente alguna información relevante, como sus nombres y detalles.