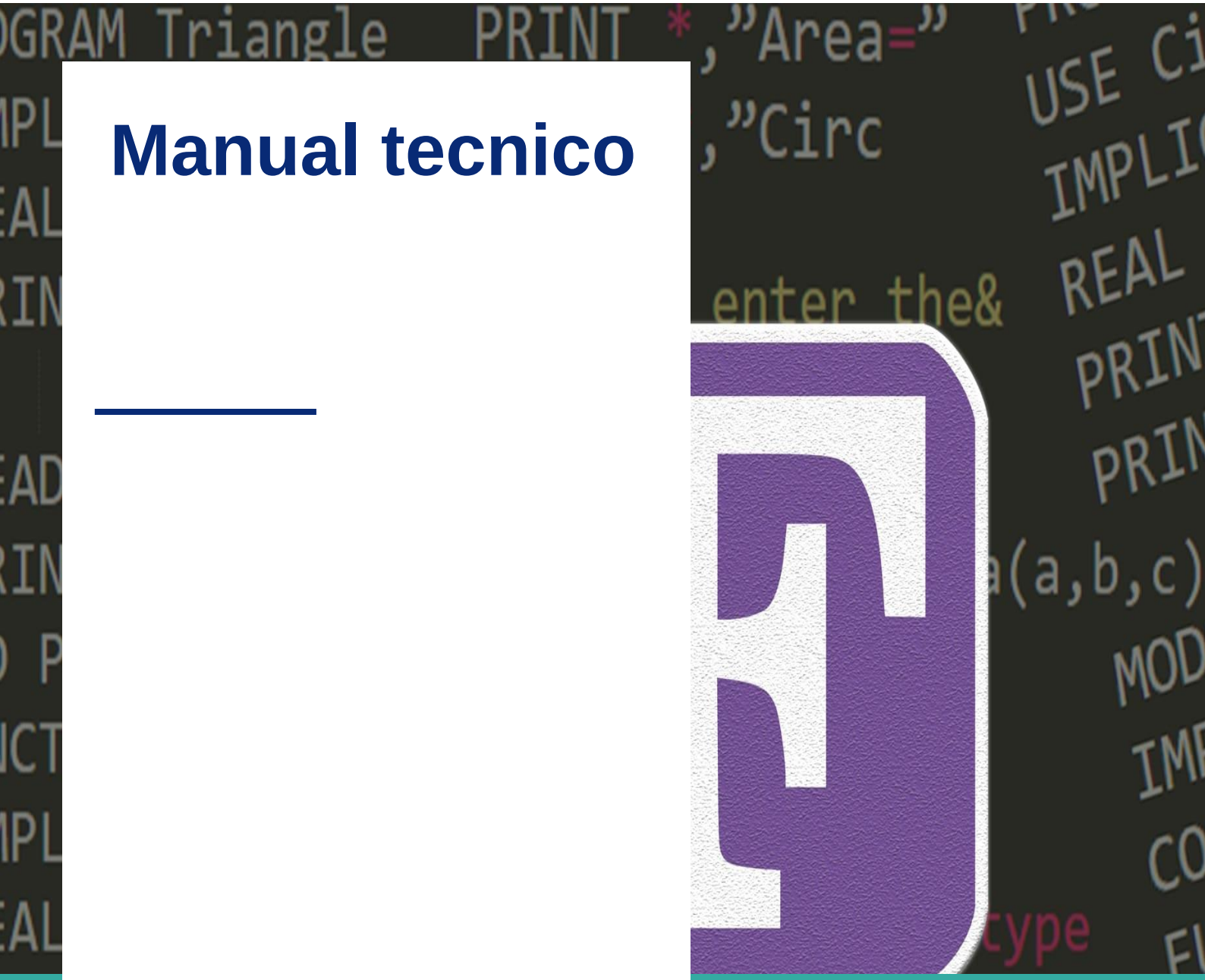
Manual tecnico



10 ABRIL

Laboratorio Estructuras de Datos
Creado por: Juan Pablo Samayoa Ruiz

USAC TRICENTENARIA
Universidad de San Carlos de Guatemala

Proyecto Fase 2

Definiciones y herramientas a utilizar

Lenguaje: Fortran 90

Sistema operativo: Windows o Linux (A criterio del usuario)

Herramientas y/o librerías: uuid_module, Json_Fortran

IDE: Visual Studio Code

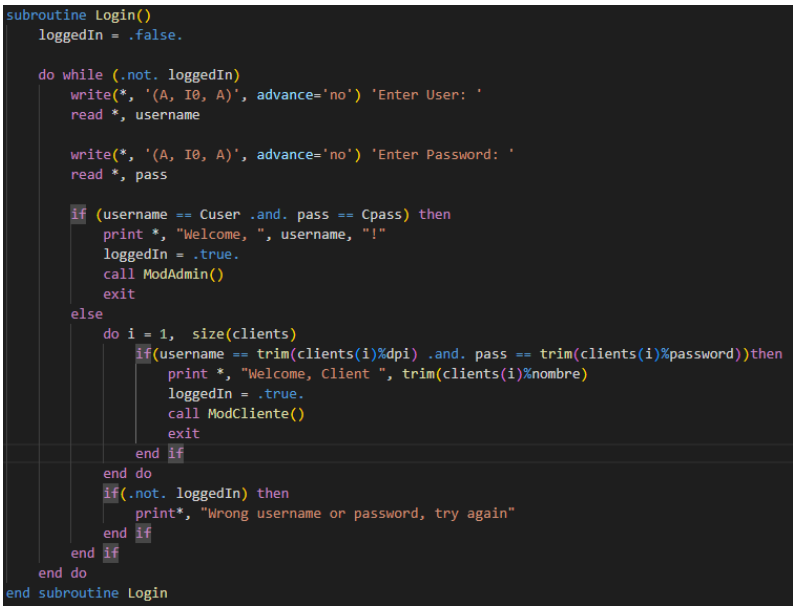
Descripcion del proyecto: Se desarrolla una aplicación en consola que pueda ejecutarse indepentedientemente del sistema operativo de ls PC, por lo que se propone crear una aplicación de consola el el lenguaje de programacion Fortran,esta aplicación debe permitir a los clientes de la empresa Pixel Print Studio registrar imágenes especiales contruidas por capas. Para poder hacer uso de la aplicación el cliente debe registrarse

La principal funcionalidad de la aplicación consiste en un generador de imágenes por capas, la aplicación cotara con un conjunto de capas cargadas previamente y almacenadas en memoria para se utilizadas, estas capas se utilizaran para generar imágenes hechas con

pixeles, cada capa contendra la informacion de los distintos pixeles y al colocar una capa sobre otra esta iran formando una imagen mas completa.

El sistema es capaz de generar imágenes seleccionando las capas deseadas.

Registro y Login:



```
subroutine Login()
    loggedIn = .false.

    do while (.not. loggedIn)
        write(*, '(A, I0, A)', advance='no') 'Enter User: '
        read *, username

        write(*, '(A, I0, A)', advance='no') 'Enter Password: '
        read *, pass

        if (username == Cuser .and. pass == Cpass) then
            print *, "Welcome, ", username, "!"
            loggedIn = .true.
            call ModAdmin()
            exit
        else
            do i = 1,  size(clients)
                if(username == trim(clients(i)%dpi) .and. pass == trim(clients(i)%password))then
                    print *, "Welcome, Client ", trim(clients(i)%nombre)
                    loggedIn = .true.
                    call ModCliente()
                    exit
                end if
            end do
            if(.not. loggedIn) then
                print*, "Wrong username or password, try again"
            end if
        end if
    end do
end subroutine Login
```

la subrutina Login funciona como un portero, permitiendo o denegando el acceso a los usuarios basándose en si pueden proporcionar las credenciales correctas. Cuando un usuario intenta iniciar sesión, se le pide que introduzca su nombre de usuario y su contraseña. Estas credenciales se comparan con las almacenadas en el sistema.
Si las credenciales del usuario coinciden con las del administrador, se le da la bienvenida y se le permite acceder al sistema como administrador. En este caso, la subrutina ModAdmin se ejecuta, lo que probablemente proporciona al usuario privilegios de administrador.
Si las credenciales del usuario no coinciden con las del administrador, el sistema comprueba si coinciden con las de cualquier cliente en una lista de clientes. Si es así, se le da la bienvenida al usuario y se le permite acceder al sistema como cliente. En este caso, se ejecuta la subrutina ModCliente, que probablemente proporciona al usuario privilegios de cliente.

Si las credenciales del usuario no coinciden con las del administrador ni con las de ningún cliente en la lista, se le informa de que el nombre de usuario o la contraseña son incorrectos y se le pide que lo intente de nuevo. Este proceso se repite hasta que el usuario proporciona las credenciales correctas y puede iniciar sesión correctamente.

Estructuras a utilizar:

-Lista circular (álbum): Estructura de datos lineal similar a una lista enlazada o doblemente enlazada, pero con una característica adicional, no tiene fin, esto se debe al puntero que apunta hacia el primer nodo desde el nodo final, haciendo que su puntero nunca apunte a NULL

-Arbol ABB: Un ABB, o Árbol Binario de Búsqueda (Binary Search Tree en inglés), es una estructura de datos de árbol que tiene la propiedad especial de que cada nodo puede tener a lo sumo dos hijos, y todos los nodos a la izquierda de un nodo tienen valores menores, mientras que todos los nodos a la derecha tienen valores mayores.

-Arbol B: Es una estructura de datos en forma de árbol que se utiliza para almacenar y recuperar datos de forma eficiente. Se caracteriza por tener nodos que pueden tener un número variable de hijos, lo que permite que el árbol se mantenga balanceado y que las operaciones de búsqueda, inserción y eliminación sean eficientes

-Arbol AVL: Es un tipo especial de árbol binario de búsqueda que se auto-balancea. Esto significa que el árbol se mantiene en un estado de equilibrio después de cada operación de inserción o eliminación, lo que garantiza que las operaciones de búsqueda sean siempre eficientes.

-Matriz dispersa: Es una matriz de gran tamaño en la que la mayor parte de sus elementos son cero. Esto significa que la mayoría de la información en la matriz está vacía, lo que puede ser un gran desperdicio de espacio de almacenamiento y tiempo de procesamiento si se utiliza una estructura de datos tradicional para almacenarla.

Archivo Json: Es un formato de intercambio de datos ligero y fácil de leer. Se basa en la sintaxis de objetos de JavaScript, pero es independiente del lenguaje, lo que significa que puede ser utilizado por cualquier lenguaje de programación.