Manual Técnico

1. Introducción

1.1 Objetivo del Sistema

El Sistema de Atención al Cliente tiene como objetivo principal simular la gestión de clientes y ventanillas, proporcionando una herramienta para evaluar la eficiencia del proceso de atención.

1.2 Tecnologías Utilizadas

El sistema está implementado en Fortran, haciendo uso de módulos para organizar y reutilizar código de manera eficiente.

1.3 Estructura del Código

El código se organiza en módulos, cada uno encargado de aspectos específicos del sistema, como la gestión de clientes, ventanillas y pilas de imágenes.

2. Módulo ColaClientes

2.1 Cliente Type

```
type, public :: Cliente
    integer :: id
    character(50) :: nombre
    integer :: img_g
    integer :: img_p
end type Cliente
```

La estructura Cliente almacena información sobre un cliente, incluyendo un identificador, nombre y contadores de imágenes.

2.2 NodoCliente Type

```
type, public :: NodoCliente
    type(Cliente) :: cliente
    type(NodoCliente), pointer :: siguiente
end type NodoCliente
```

El NodoCliente representa un nodo en la cola de clientes, conteniendo un cliente y un puntero al siguiente nodo.

2.3 ColaClientes Module Subroutines

El módulo ColaClientes contiene subrutinas para cargar clientes desde archivos JSON y otras funciones relacionadas con la gestión de clientes.

3. Módulo VentanillaModule

3.1 NodoPilaType

```
type, public :: NodoPilaType
    integer :: idImagen
    type(NodoPilaType), pointer :: siguiente => null()
end type NodoPilaType
```

La estructura NodoPilaType representa un nodo en la pila de imágenes de una ventanilla.

3.2 PilaType

```
type, public :: PilaType
    type(NodoPilaType), pointer :: tope => NULL()
end type PilaType
```

La estructura PilaType implementa una pila de imágenes mediante un puntero al tope de la pila.

3.3 VentanillaType

```
type, public :: VentanillaType
    integer :: id
    type(Cliente) :: Clientes
    logical :: disponible = .true.
    type(PilaType) :: pila
    type(VentanillaType), pointer :: siguiente => null()
end type VentanillaType
```

La estructura VentanillaType representa una ventanilla con un identificador, cliente asignado, disponibilidad y una pila de imágenes.

3.4 listaVentanillas Type

```
type, public :: listaVentanillas
    type(VentanillaType), pointer :: tope => NULL()
end type listaVentanillas
```

La estructura listaVentanillas mantiene un puntero al tope de la lista de ventanillas.

3.5 VentanillaModule Subroutines

El módulo VentanillaModule contiene subrutinas para la inicialización de pilas, apilar y desapilar imágenes, la creación de ventanillas y la búsqueda de ventanillas disponibles.

4. Módulo PilaImg

4.1 PilaImg Module Subroutines

El módulo PilaImg proporciona subrutinas para llamar a subrutinas específicas según el tipo de imagen y crear nodos de ventanilla.

5. Programa Principal - MainProgram

5.1 Menú Principal

El programa principal MainProgram contiene un menú interactivo con opciones para la carga masiva de clientes, configuración de ventanillas, ejecución de pasos, consulta de estado y generación de reportes.

5.2 Subrutinas del Menú Principal

Las subrutinas dentro del MainProgram gestionan las diferentes opciones del menú, interactuando con los módulos correspondientes.

5.3 Interacción con Módulos

El programa principal interactúa con los módulos ColaClientes, VentanillaModule, y PilaImg para realizar funciones específicas como la carga de clientes, gestión de ventanillas y pilas de imágenes.

6. Consideraciones de Implementación

6.1 Carga Masiva de Clientes

La carga masiva de clientes se realiza a través del módulo ColaClientes, que lee información desde archivos JSON y carga los clientes en una cola.

6.2 Configuración de Ventanillas

El número de ventanillas se configura mediante el módulo VentanillaModule, que crea una lista enlazada de ventanillas.

6.3 Ejecución de Pasos

La ejecución de pasos simula el proceso de atención, asignando clientes a ventanillas disponibles y gestionando las pilas de imágenes.

6.4 Estado en Memoria

La consulta del estado en memoria a través del menú principal proporciona información detallada sobre clientes, ventanillas y pilas de imágenes.

6.5 Generación de Reportes

La generación de reportes se realiza a través de subrutinas específicas que recopilan datos relevantes sobre la eficiencia del sistema.

7. Dependencias Externas

7.1 JSON_Module

El módulo ColaClientes hace uso de la dependencia externa JSON_Module para la carga y manipulación de datos almacenados **en archivos JSON.**