

PROYECTO

PIDE COLA

Integrante:

Froilan Roa.

Profesor: Joselito De Sousa.

El proyecto consta de:

(1) .cpp:

1- pro.cpp:

Este es el programa principal, donde esta el menú que llama a todas las funciones.

(3) .h

1- Tda.h:

Están todas las estructuras definidas, todas las funciones del programa y al final la lectura y escritura de los archivos de texto.

2- Usuario.h:

Están las estructuras de los datos que tienen las estructuras definidas en "Tda.h"

3- Fecha.h:

Esta la función que verifica si el día ha cambiado.

(6) .txt:

1- lista.txt:

En este archivo de almacena la información de todos los usuarios registrados, el programa cuando se inicie los cargara de la misma.

2- listacarros.txt:

En este archivo se almacena la información de todos los carros registrados, el programa cuando se inicie los cargara de la misma.

3- listacolas.txt:

En este archivo se almacena la información de las solicitudes de cola que se hayan hecho, el programa cuando se inicie las cargara de la misma.

4-listadestinos.txt:

En este archivo se almacenan los destinos de los lugares a donde los usuarios han solicitado la cola, el programa cuando se inicie las cargara de la misma.

5-listaofrece.txt:

En este archivo se almacenan todas las peticiones de cola que hayan hecho los usuarios, el programa cuando se inicie las cargara de la misma.

6-dia.txt:

En este archivo se almacena el día en el cual el programa fue cerrado, al iniciarse verificara si es el mismo, de dicha lista, y cargara las solicitudes y ofrecimientos de colas dependiendo del día.

ESTRUCTURAS:

Definidas en: "Tda.h":

1- listaUsuarios:

Aquí están los campos para la información de todos los usuarios que se van a registrar (nodo: definido en "Usuarios.h") y el puntero hacia el próximo elemento del mismo tipo.

2- listaCarros:

Aquí están los campos para la información de todos los carros de los usuarios (carro: definido en "Usuarios.h") y el puntero hacia el próximo elemento del mismo tipo.

3- listaColas:

Aquí están los campos para la información de todos los usuarios que pidieron la cola (cola: definido en "Usuarios.h") y el puntero hacia el próximo elemento del mismo tipo.

4- listaOfrece:

Aquí están los campos para la información de todos los usuarios que ofrecieron la cola (ofrece: definido en "Usuarios.h") y el puntero hacia el próximo elemento del mismo tipo.

5- listaDestinos:

Aquí están los campos para la información de todos los destinos de los usuarios que pidieron el servicio de colas (destino: definido en "Usuarios.h") y el puntero hacia el próximo elemento del mismo tipo.

6- listaProcesa:

Aquí están los campos para la información de todas las colas procesadas, se guarda el numero de cédula de las personas que hacen Mach al solicitar las colas.

FUNCIONES:

1- void registrarPersona(Tlista &lista, Tcarro &listacarros, int cedula):

verifica si la persona esta registrada y si tiene carro, llamando a otra función que crea el nodo y depuse lo inserta en la lista.

2- Tlista crearNodo(Tlista &lista, int cedula, bool posee);

Crea el nodo y le pide la información a insertar del usuario nuevo a registrar.

3- bool esVaciaLista(Tlista &lista):

Verifica si la lista de usuarios esta vacía.

4- void modificar(Tlista &lista, Tcarro &listacarros, int cedula):

Modifica los datos de un usuario.

5- void mostrarModificacion(Tlista variable):

Muestra los datos de la persona ha modificar.

6- void registrarCarro(Tcarro &listacarros, int cedula):
registra los carros de los usuarios, llamando a otra función que crear el nodo de los carros y los inserta en la lista.

7- Tcarro crearCarro(Tcarro &listacarros, int cedula):
crea el nodo y pide la información del mismo.

8- bool esVaciaCarros(Tcarro &listacarros):
verifica si la lista de los carros esta vacía.

9- void eliminarCarro(Tcarro &listacarros, int valor):
elimina un carro que esta registrado de la lista.

10- void pedirCola(Tlista &lista, Tcolas &listacolas, Tdestino &listadestinos, int cedula):
verifica si el usuario introducido existe y llama a otra función que registra en la lista la solicitud de cola del mismo usuario.

11- void registrarCola(Tlista variable, Tcolas &listacolas, Tdestino &listadestinos):
llama a una función que crea el nodo de la solicitud de cola del usuario y lo registra en la lista.

12- void registrarColaModificada(Tcolas nodo , Tcolas &listacolas):
Esta función registra las colas modificadas cuando ya la solicitud ha sido procesada, es decir no pide ningún dato, solo recibe el nodo a insertar.

13 - bool verificarDestinoRepetido(char destino[100], Tdestino &listadestinos):
Esta función verifica si un destino a insertar esta repetido, para que no se repitan en la lista al momento de insertar nuevos destinos.

14- Tcolas crearNodoCola(Tlista variable, Tcolas &listacolas, Tdestino &listadestinos):
crea el nodo de las personas que piden la cola y solicita la información de la misma.

15- void registrarDestinos (char destino[100], Tdestino &listadestinos):
registra el destino que se le esta pasando a la lista de destinos, que es donde los usuarios que van a ofrecer la cola usaran al momento de ofrecer el servicio.

16- Tdestino crearDestino(Tdestino &listadestinos, char destino[100]):
crea el nodo del destino que se va a insertar en la lista de los destinos.

17- bool buscarDireccionClave(Tdestino &listadestinos, int codigo):
verifica si existe el código introducido por el usuario al momento de escoger un destino para ofrecer la cola.

18- void eliminarColaModificada(Tcolas &listacolas, int valor):
Esta función elimina una una petición de cola si esta procesada para posteriormente insertar una nueva al final de la lista.

19- void mostrarCola(Tlista variable):
muestra la solicitud de cola si no esta procesada.

20- void modificarCola(Tcolas &listacolas, Tdestino &listadestinos, int cedula, Tprocesa &listaprocesa) :

Modifica la solicitud de cola introducida, verificando si esta no ha sido procesada, para eliminarla y crear una nueva o simplemente modificarla.

21- void mostrarModificacionCola(Tcolas variable):

muestra los datos de la solicitud de cola mientras el usuario la esta modificando.

22- void eliminarCola(Tcolas &listacolas, int valor):

elimina una solicitud de cola.

23- void mostrarSolicitudProcesada(Tcolas variable, Tlista &lista, Tcarro &listacarros, int cedula):

Si la solicitud esta procesada muestra los datos del usuario que va a prestarle es servicio de colas a el usuario que lo solicito.

24- void mostrarSolicitudCola(Tcolas &listacolas, int cedula, Tprocesa &listaprocesa, Tcarro &listacarros, Tlista &lista):

verifica si la solicitud de cola fue procesada y llama a una función distinta en el caso que sea, es decir, una función para mostrar la solicitud procesada y otra para mostrar la solicitud no procesada.

25- bool esVaciaColas(Tcolas &listacolas):

verifica si la lista de solicitudes de colas esta vacía.

26- void mostrarSolicitudDestino(Tcolas &listacolas, Tdestino &listadestinos):

muestra las solicitudes de cola por destinos puntuales.

27- void ofrecerCola(Tlista &lista, Tofrece &listaofrece, Tdestino &listadestinos, int cedula):

verifica que el usuario a ofrecer la cola exista, si existe llama a otra función que registra en la lista los ofrecimientos de colas.

28- void registrarOfrece(Tlista variable, Tofrece &listaofrece, Tdestino &listadestinos):

inserta en la lista de ofrecimientos las colas ofrecidas.

29- void mostrarDestinos(Tdestino variable):

muestra los destinos de las solicitudes de colas hechas por los usuarios.

30- Tofrece crearNodoOfrece(Tlista variable, Tofrece &listaofrece, Tdestino &listadestinos):

crea el nodo a insertar en la lista de los ofrecimientos de cola pidiendo la información que se va registrar.

31- void mostrarOfrecimientosModificacion(Tofrece variable):

muestra la información que tiene el ofrecimiento de cola que se va a modificar.

- 32- void modificarOfrece(Tofrece &listaofrece, int cedula):
modifica los ofrecimientos de cola que el usuario hizo.
- 33- void mostrarOfrecimientos(Tofrece variable):
muestra todos los ofrecimientos de cola que han hecho los usuarios.
- 34- void eliminarOfrece(Tofrece &listaofrece, int valor):
elimina el ofrecimiento de cola hecho por algún usuario.
- 35- bool esVaciaOfrece(Tofrece &listaofrece):
verifica si la lista de los ofrecimientos esta vacía.
- 36- void escribirLista(Tlista variable):
Escribe en el fichero lista.txt la información de los usuarios registrados.
- 37- void escribirListaCarros(Tcarro variable):
Escribe en el fichero listacarros.txt la información de los carros registrados.
- 38- void escribirListaDestino(Tdestino variable):
Escribe en el fichero listadestinos.txt la información de los destinos registrados.
- 39- void cargarPersona(Tlista nodo, Tlista &lista):
carga en la lista de las personas los nodos que les pasa la función que lee los
ficheros.
- 40- void cargarCarro(Tcarro nodocarro, Tcarro &listacarros):
carga en la lista de los carros los nodos que les pasa la función que lee los
ficheros.
- 41- void cargarCola(Tcolas nodo, Tcolas &listacolas):
carga en la lista de las solicitudes de colas los nodos que les pasa la función
que lee los ficheros.
- 42- void cargarDestino (Tdestino nodo, Tdestino &listadestinos):
carga en la lista de los destinos de colas los nodos que les pasa la función que
lee los ficheros.
- 43- void cargarOfrece(Tofrece nodo, Tofrece &listaofrece):
carga en la lista de los ofrecimientos de colas los nodos que les pasa la función
que lee los ficheros.
- 44- void cargarLista(Tlista &lista, char dirLista[200]):
carga los nodos de los usuarios que estan en los ficheros y luego se la pasa a la
función que los registra en la respectiva lista.
- 45- void cargarListaCarros(Tcarro &listacarros, char dirListaCarros[200]):
carga los nodos de los carros que estan en los ficheros y luego se la pasa a la
función que los registra en la respectiva lista.
- 46- void cargarListaColas(Tcolas &listacolas, char dirListaCarros[200]):
carga los nodos de las peticiones de cola que estan en los ficheros y luego se la
pasa a la función que los registra en la respectiva lista.

47- void cargarListaDestinos(Tdestino &listadestinos, char dirListaDestinos[200]):
carga los nodos de los destinos de las peticiones de cola que estan en los
ficheros y luego se la pasa a la función que los registra en la respectiva lista.

48- void cargarListaOfrecimientos(Tofrece &listaofrece, char dirListaOfrecimientos[200]):
carga los nodos de los ofrecimientos de cola que estan en los ficheros y luego
se la pasa a la función que los registra en la respectiva lista.

LIBRERIAS:

- 1- <iostream>:
para las operaciones de entrada y salida de datos.
- 2- <string>, <string.h>:
para el manejo de las cadenas de caracteres.
- 3- <stdio.h>:
para el manejo de algunas constante en el programa.
- 4- <fstream>:
para el manejo de ficheros, lectura y escritura.
- 5- <stdlib.h>:
para la reserva de memoria (malloc).
- 6- <ctype.h>:
Para poner en mayúscula todos los strings introducidos.