

Universidad de San Carlos de Guatemala
Facultad de ingeniería
Estructura de datos
Ing. Luis Espino
Aux. Robinson Perez
Diciembre 2017

Practica 1

Manual técnico

Paul Steve Contreras Herrera
201408489

Definición de archivos

- x Archivos.h
 - ➔ customview.h
 - ➔ estructuras.h
 - ➔ mainwindow.h

- x Archivos.cpp
 - ➔ customview.cpp
 - ➔ estructuras.cpp
 - ➔ mainwindow.cpp
 - ➔ main.cpp

- x Formularios
 - ➔ mainwindow.ui

Definición de struct

----- Nodos de las listas -----

```
struct NodoAvion
{
    NodoAvion *anterior;           / puntero de nodo
    NodoAvion *siguiente;         / puntero de nodo
    Desbordaje *desbordaje;       / puntero de lista
    void insertarPasajeros();      / metodo de agregar aviones
    NodoAvion();                  / constructor
};

struct NodoPasajero
{
    int idPasajero;                / atributo
    int noMaletas;                 / atributo
    int noDocumentos;             / atributo
    int noTurnos;                 / atributo
    NodoPasajero *siguiente;      / puntero de nodo
    NodoPasajero();              / constructor
};

struct NodoDocumento
{
    int idPasajero;                / atributo
    int numero;                   / atributo
    NodoDocumento *siguiente;     / puntero de nodo
    NodoDocumento();             / constructor
};

struct NodoEscritorio
{
    string id;                    / atributo
    int capacidad;                / atributo
    int idCliente;                / atributo
    string estado;                / atributo
    int documentos;              / atributo
    int turnosRestantes;          / atributo
    NodoPasajero *pPasajero;      / puntero de nodo
    NodoDocumento *topDocumento; / puntero de nodo
    NodoEscritorio *anterior;     / puntero de nodo
    NodoEscritorio *siguiente;    / puntero de nodo
    NodoEscritorio(string ide);   / constructor
};

struct NodoMaleta
{
    int idMaleta;                 / atributo
    int numero;                   / atributo
    NodoMaleta *siguiente;        / puntero de nodo
    NodoMaleta *anterior;         / puntero de nodo
    NodoMaleta(int id, int num);  / constructor
};
```

```

struct NodoServicio
{
    int idAvion;           / atributo
    int idEstacion;       / atributo
    string estado;        / atributo
    int turnosRestantes;  / atributo
    NodoServicio *siguiente; / puntero de nodo
    NodoServicio(int id); / constructor
};

struct NodoCola
{
    int idAvion;           / atributo
    int turnosServicio;    / atributo
    NodoCola *siguiente;  / puntero de nodo
    NodoCola(int id, int turnos); / constructor
};

```

----- lista de aviones -----

```

struct LlegadaAviones
{
    NodoAvion *primero;      puntero de nodo
    NodoAvion *ultimo;      / puntero de nodo
    int tipo;               / atributo
    int pasajeros;          / atributo
    int mante;              / atributo
public:
    LlegadaAviones();       / constructor
    void eliminar();        / elimina un nodo de la lista
    void insertar(int total); / inserta un valor en la lista
    int random(int min, int max); / genera un numero random
    NodoAvion *crearAvion(int id); / constructor
};

```

----- lista de aviones -----

```

struct Desbordaje
{
private:
    int maletas;           / atributo
    int documentos;       / atributo
    int turnos;           / atributo
public:
    int totalMaletas;      / atributo
    NodoPasajero *ultimo;  / puntero de nodo
    NodoPasajero *primero; / puntero de nodo
    ListaMaletas *listaMaletas; / puntero de nodo lista
    Desbordaje();         / constructor
    void eliminar(int idp); / elimina un elemento de la lista
    void insertar(int total); / agrega un elemento a la lista
    int random(int min, int max); / genera un numero random
    NodoPasajero *crearPasajero(); / crean un nodo pasajero
};

```

----- lista de maletas -----

```
struct ListaMaletas
{
public:
    NodoMaleta *primero;           / puntero de nodo
    NodoMaleta *ultimo;           / puntero de nodo
    ListaMaletas();               / constructor
    void insertar(int id, int total); / ingresar en lista
    void eliminar(int id, int total); / elimina de la lista
};
```

----- lista de escritorios -----

```
struct Escritorios
{
public:
    NodoEscritorio *actual;        / puntero de nodo
    NodoEscritorio *ultimo;        / puntero de nodo
    NodoEscritorio *primero;       / puntero de nodo
    NodoPasajero *aPasajero;       / puntero de nodo
    Escritorios();                 / constructor
    void eliminarLista();          / elimina toda la lista
    void popDocumentos();          / saca los documentos del nodo
    void insertar(int total);
    void sacarPasajero(NodoEscritorio *eActual); / saca pasajero de la lista
    void nuevoRegistro(NodoEscritorio *ventana); / lo guarda en el nodo
    void pushDocumentos();         / ingresa los docs a la pila
    bool encolarPasajeros(NodoPasajero *pasajero); / mueve los pasajeros a cola
private:
    string nuevoId(int val);       / auxiliar para ids
};
```

----- lista de mantenimiento -----

```
struct Mantenimiento
{
public:
    NodoServicio *primero;         / puntero de nodo
    NodoServicio *ultimo;         / puntero de nodo
    NodoServicio *actual;         / puntero de nodo
    Mantenimiento();              / constructor
    void eliminarLista();          / eliminar toda la lista
    void insertar(int total);      / inserta en la lista
    void eliminarAvion(int idAvion); / elimina en lista
    void ingresarAvion(int idAvion, int turnos); /ingresa en lista
};
```