Variables

Las variables son espacios reservados en la memoria que, como su nombre indica, pueden cambiar de contenido a lo largo de la ejecución de un programa. Una variable corresponde a un área reservada en la memoria principal del ordenador.

Para que nuestro código sea más entendible y claro, el identificador de la variable debe auxiliar a la memoria, es decir que debe reflejar el uso dentro del programa de la misma.

```
//Variables
int turnos=0;
int contaviones=0;
int contpasajeros=0;
int contmaletas=0;
int contescritorios=64;
int contarestaciones=0;
int contadocs=0;
```

Estructuras de Datos

En ciencias de la computación, una estructura de datos es una forma particular de organizar datos en una computadora para que pueda ser utilizado de manera eficiente.

Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.

Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet. Por lo general, las estructuras de datos eficientes son clave para diseñar algoritmos eficientes. Algunos métodos formales de diseño y lenguajes de programación destacan las estructuras de datos, en lugar de los algoritmos, como el factor clave de organización en el diseño de software.

```
struct Avion
{
    QString nombre;
    int tipo;
    int pasajero;
    int desbordaje;
    int mantenimiento;

    struct Avion *siguiente;
    struct Avion *anterior;
}*avionprimero,*avionultimo=NULL;
struct Pasajero{
    QString nombre;
    int maleta;
    int documento;
    int registro;

    struct Pasajero *siguiente;
}*pasajeroprimero,*pasajeroultimo=NULL;
```

```
struct Maleta
{
    QString nombre;
    Maleta *siguiente;
    Maleta *anterior;
}*maletaprimero,*maletaultimo;
struct  Persona{
    QString nombre;
    int turnos;
    int  maletas;
    int docs;
    Persona *siguiente;
    Persona *anterior;
};

struct Documentos{
    QString nombre;
    Documentos *siguiente;
};
struct Escritorio{
    QString nombre;
    Persona *primerapersona;
    Persona *ultimapersona;
    Documentos *docprimero;
    Documentos *docultimo;
    Escritorio *siguiente;
    Escritorio *anterior;
    int libre;
}*escritorioprimero,*escritorioultimo;
struct ColaAvion{
    QString nombre;
    int turnos;
    ColaAvion *siguiente;
}*primeravionencola,*ultimoavionencola;
struct Mantenimiento{
    QString nombre;
    QString avion;
    int turnoslibre;
    int libre;
    Mantenimiento *siguiente;
}*mantprimero,*mantultimo;
```

Métodos

Una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia. Análogamente a los procedimientos en lenguajes imperativos, un método consiste generalmente de una serie de sentencias para llevar a cabo una acción, un juego de parámetros de entrada que regularán dicha acción o, posiblemente, un valor de salida (o valor de retorno) de algún tipo.

```
void DatosAvion(int tipo,int pasajero,int desbordaje, int mantenimiento){ ...}
void CrearAviones(){ ...}

void MostrarAvion(){ ...}
void EliminarAvion(){ ...}
}
//Pasajeros

void NuevoPasajero(){ ...}
}
void MostrarPasajeros(){ ...}
}
void EliminarPasajero(){ ...}

void IngresarMaleta(){ ...}

void MostrarMaletas(){ ...}
void EliminarMaletas(){ ...}

void IngresarDocs(Escritorio *escritorio){ ...}
void EliminarDocs(Escritorio *escritorio){ ...}
}
void InsetarPasajerosAlaCola(int turns,QString nombre,int  maletas,int docs){ ...}

void CrearEscritorios(){ ...}
}
void ColaEliminarPersona(Escritorio *escritorio){ ...}
}
QString grpah;

//Mantenimiento


void CrearEstacionesMantenimiento(){ ...}

void InsertarColaAvion(QString nombre,int turnos){ ...}


void ColaEliminarAvion(){ ...}
```

Graphviz

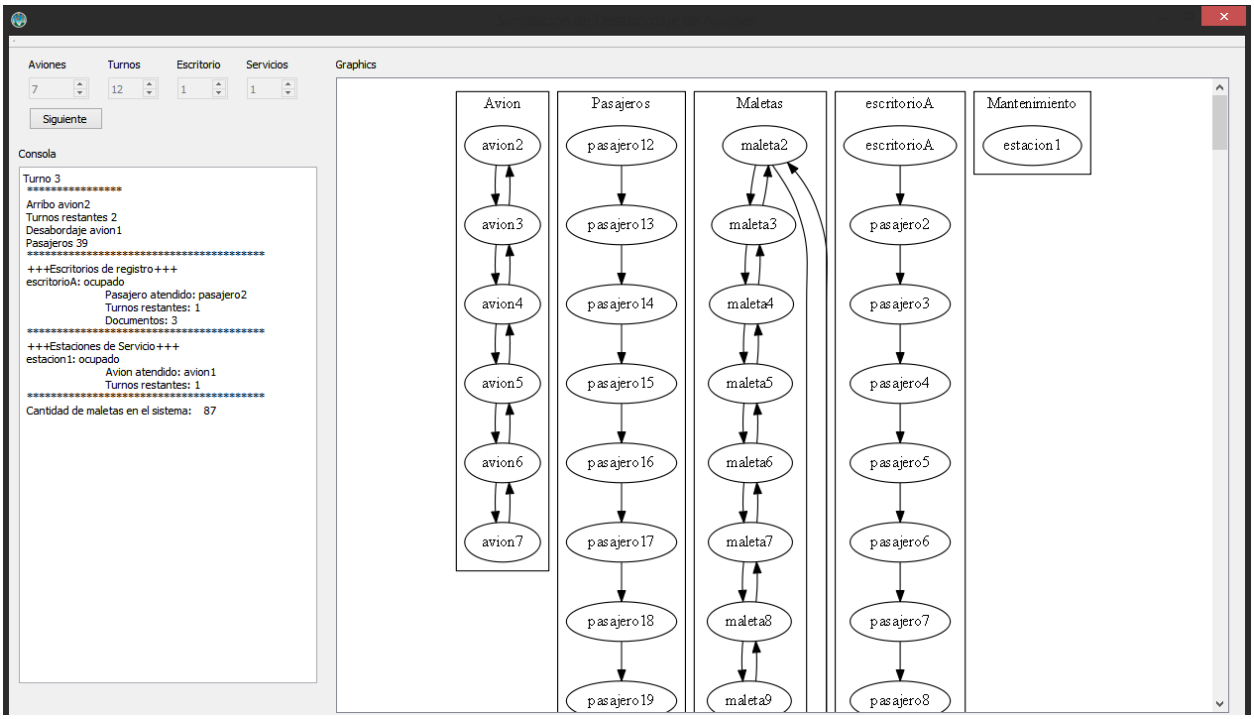
Es una aplicación de visualización de gráficos de código abierto que incluye un gran número de programas de trazado de grafico; además cuenta con interfaces interactivas y vía web, así como herramientas auxiliares y bibliotecas de funciones, existiendo versiones tanto para Windows como para Linux. Los programas de diseño de Graphviz parten de descripciones de gráficos en texto plano, lo que les permite ser editados por los usuarios y no necesitar un programa adicional para ello.

Es un conjunto de herramientas de software para el diseño de diagramas definido en el lenguaje descriptivo. Provee una forma simple de describir gráficas que sean entendibles por humanos y computadoras. Los programas en DOT generalmente usan la extensión *.gv* o *.dot*.

```
StringDotEditor=QString("digraph G { dpi=%1; ").arg(tam);
//Avion
aviontext=("subgraph clusterAviones{label = Avion;");
avion=avionprimero;
while(avion!=NULL){
    if(avion->siguiente==NULL){aviontext=QString("%1 %2;").arg(aviontext).arg(avion->nombre);}
    else{
        aviontext=QString("%1 %2->%3;").arg(aviontext).arg(avion->nombre).arg(avion->siguiente->nombre);
        aviontext=QString("%1 %2->%3;").arg(aviontext).arg(avion->siguiente->nombre).arg(avion->nombre);
    }
    avion=avion->siguiente;
}
aviontext=QString("%1}").arg(aviontext);
//Pasajero
pasajerotext=("subgraph clusterPasajero{label=Pasajeros;");
pasajero=pasajeroprimero;
while(pasajero!=NULL){
    if(pasajero->siguiente==NULL){pasajerotext=QString("%1 %2;").arg(pasajerotext).arg(pasajero->nombre);}
    else{
        pasajerotext=QString("%1 %2->%3;").arg(pasajerotext).arg(pasajero->nombre).arg(pasajero->siguiente->nombre);
    }
   colatext=QString("%1}").arg(colatext);
   //Maletas
   maletatext=("subgraph clusterMaleta{ label=Maletas");
   maleta=maletaprimero;
   if(maleta!=NULL){
       do{
           if(maleta->siguiente!=NULL){
               maletatext=QString("%1 %2->%3;").arg(maletatext).arg(maleta->nombre).arg(maleta->siguiente->nombre);
               maletatext=QString("%1 %2->%3;").arg(maletatext).arg(maleta->siguiente->nombre).arg(maleta->nombre);
               maleta=maleta->siguiente;
           }
       }while(maleta!=maletaprimero);

   }
   maletatext=QString("%1}").arg(maletatext);
   StringDotEditor=QString("%1 %2 %3 %4 %5 %6 %7 %8}").arg(StringDotEditor).arg(aviontext).arg(pasajerotext).arg(maleta
   StringDotEditor=StringDotEditor.remove("\"");
   return StringDotEditor;
}


ofstream fs("C:\\Users\\usuario\\Documents\\USAC\\EDD\\practica\\Graph\\dot.dot ");
fs<<DotEditor(tam).toStdString()<<endl;
fs.close();
system("dot -Tpng C:\\Users\\usuario\\Documents\\USAC\\EDD\\practica\\Graph\\dot.dot -o C:\\Users\\usuario\\I
imageObject = new QImage();
imageObject->load("C:\\Users\\usuario\\Documents\\USAC\\EDD\\practica\\Graph\\dot.png");
image = QPixmap::fromImage(*imageObject);
scene = new QGraphicsScene(this);
scene->addPixmap(image);
scene->setSceneRect(image.rect());
ui->graphicsView->setScene(scene);
```


Aviones
Turnos
Escritorio
Servicios
7
12
1
1
Siguiente
Consola
Turno 3
****
Arribo avion2
Turnos restantes 2
Desabordaje avion1
Pasajeros 39
****
+++Escritorios de registro+++
escritorioA: ocupado
Pasajero atendido: pasajero2
Turnos restantes: 1
Documentos: 3
****
+++Estaciones de Servicio+++
estacion1: ocupado
Avion atendido: avion1
Turnos restantes: 1
****
Cantidad de maletas en el sistema: 87
Graphics
Avion
avion2
avion3
avion4
avion5
avion6
avion7
Pasajeros
pasajero12
pasajero13
pasajero14
pasajero15
pasajero16
pasajero17
pasajero18
pasajero19
Maletas
maleta2
maleta3
maleta4
maleta5
maleta6
maleta7
maleta8
maleta9
escritorioA
escritorioA
pasajero2
pasajero3
pasajero4
pasajero5
pasajero6
pasajero7
pasajero8
Mantenimiento
estacion1