Linea= **Clase** de

nombre: cadena

fecha\_origen: fecha

**Método** llenar ()

**Inicio**

Escribir (“nombre de la línea”)

leer nombre

fecha\_origen←nuevo (Fecha)

fecha\_origen.llenar()

**Fin Método**

**Fin Clase**

Fecha= **Clase** de

dd,mm,aaaa:entero

**Método** llenar()

**Inicio**

escribir (“ingrese el año”)

leer aaaa

mientras (aaaa<=0) hacer

escribir (“El año debe ser mayor que 0. Ingréselo nuevamente”)

leer aaaa

fin mientras

escribir (“ingrese el mes”)

leer mm

mientras( (mm<=0) O (mm>12) ) hacer

escribir (“El mes debe oscilar entre 1 y 12. Ingréselo nuevamente”)

leer mm

fin mientras

escribir (“ingrese el dia”)

leer dd

si ( ( mm=1) O ( mm=3) O ( mm=5) O ( mm=7) O ( mm=8) O ( mm=10) O (

mm=12) ) entonces

mientras ( (dd<=0) O (dd>31)) hacer

escribir (“El día debe oscilar entre 1 y 31. Introduzca

nuevamente el valor”)

leer dd

fin mientras

sino si( ( mm=4) O ( mm=6) O ( mm=9) O ( mm=11) ) entonces

mientras ( (dd<=0) O (dd>30)) hacer

escribir (“El día debe oscilar entre 1 y 31. Introduzca

nuevamente el valor”)

leer dd

fin mientras

sino

mientras ( (aaaa%4=0) Y ((dd<=0) O (dd>29)) ) hacer

escribir (“El día debe oscilar entre 1 y 29. Introduzca

nuevamente el valor”)

leer dd

fin mientras

mientras ( (aaaa%4<>0) Y ((dd<=0) O (dd>28)) ) hacer

escribir (“El día debe oscilar entre 1 y 28. Introduzca

nuevamente el valor”)

leer dd

fin mientras

fin si

**Fin Método**

**Fin Clase**

Hora= **Clase** de

hh.mm: entero

**Método** llenar ()

**Inicio**

escribir (“Ingrese la hora (0 a 23)”)

leer hh

mientras ( (hh<0) O (hh<23) ) hacer

escribir (“La hora debe oscilar entre 0 y 23 .Ingrésela nuevamente”)

leer hh

fin mientras

escribir (“Ingrese los minutos (0 a 59)”)

leer mm

mientras ( (mm <0) O (mm <23) ) hacer

escribir (“Los minutos deben oscilar entre 0 y 59 .Ingréselos

nuevamente”)

leer mm

fin mientras

**Fin Método**

**Método** comparar (salida: hora): booleano

**Inicio**

si (salida.hh>hh) entonces

comparar←falso

sino si (salida.hh=hh) Y (salida.mm>=mm) entonces

comparar←falso

sino

comparar←verdadero

fin si

**Fin Método**

**Fin Clase**

Azafat= **Clase** de

cedula: entero

nombre, apellido: cadena

sexo: carácter

**Método** llenar ()

**Inicio**

escribir (“cédula”)

leer cedula

escribir (“nombre”)

leer nombre

escribir (“apellido”)

leer apellido

escribir (“sexo (f/m)”)

leer sexo

mientras ( (sexo<>‘f’) Y (aux<>’m’) ) hacer

escribir (“Opción incorrecta. ingrese el sexo? f/m”)

leer (sexo)

fin mientras

**Fin Método**

**Método** encontrar\_cedula (valor: entero)

**Inicio**

si (valor=cedula) entonces

encontrar\_cedula←verdadero

sino

encontrar\_cedula←falso

fin si

**Fin Método**

**Método** editar ()

**Variables**

aux: caracter

**Inicio**

escribir (“¿Desea modificar el nombre? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar el nombre? s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese el nuevo nombre:”)

leer (nombre)

fin si

escribir (“¿Desea modificar el apellido? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar el apellido? s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese el nuevo apellido:”)

leer (apellido)

fin si

escribir (“¿Desea modificar el sexo? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar el sexo? s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese el nuevo sexo:”)

leer (sexo)

mientras ( (sexo<>‘f’) Y (aux<>’m’) ) hacer

escribir (“Opción incorrecta. ingrese el sexo? f/m”)

leer (sexo)

fin mientras

fin si

**Fin Método**

**Fin Clase**

Piloto= **Clase** de

cedula: entero

nombre, apellido: cadena

fecha\_prof: fecha

**Método** llenar()

**Inicio**

escribir (“cédula”)

leer cedula

escribir (“nombre”)

leer nombre

escribir (“apellido”)

leer apellido

fecha\_prof:←nuevo (Fecha)

fecha\_prof.llenar()

**Fin Método**

**Método** encontrar\_cedula (valor: entero)

**Inicio**

si (valor=cedula) entonces

encontrar\_cedula←verdadero

sino

encontrar\_cedula←falso

fin si

**Fin Método**

**Método** editar ()

**Variables**

aux: caracter

**Inicio**

escribir (“¿Desea modificar el nombre? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar el nombre? s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese el nuevo nombre:”)

leer (nombre)

fin si

escribir (“¿Desea modificar el apellido? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar el apellido? s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese el nuevo apellido:”)

leer (apellido)

fin si

escribir (“¿Desea modificar la fecha de profesión? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar la fecha de profesión?

s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese la nueva fecha:”)

fech\_prof.llenar()

fin si

**Fin Método**

**Fin Clase**

Avion= **Clase** de

id, modelo, linea: cadena

capacidad,carga\_max: entero

**Método** llenar ()

**Inicio**

escribir (“ingrese el id”)

leer (id)

escribir (“ingrese el imodelo”)

leer (modelo)

escribir (“ingrese el nombre de la aerolínea al que corresponde”)

leer (linea)

**Fin Método**

**Método** encontrar\_linea (arrlin[]:Linea): booleano

**Variables**

i:entero

**Inicio**

para i (1 hasta 30) hacer

si (arrlin[i]<>NULO) entonces

si (arrlin[i].nombre=linea) entonces

encontrar\_linea←verdadero

fin si

fin si

fin para

encontrar\_linea←falso

**Fin Método**

**Método** encontrar\_avion (valor:cadena): booleano

**Inicio**

si (id=valor) entonces

encontrar\_avion←verdadero

sino

encontrar\_avion←falso

fin si

**Fin Método**

**Método** editar ()

**Variables**

aux: caracter

**Inicio**

escribir (“¿Desea modificar el modelo? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar el modelo? s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese el nuevo modelo:”)

leer (modelo)

fin si

escribir (“¿Desea modificar la capacidad? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar la capacidad? s/n”)

leer (aux)

fin mientras

si (aux=’s’) entonces

escribir (“Ingrese la capacidad:”)

leer (capacidad)

fin si

escribir (“¿Desea modificar la aerolínea a la que pertenece? s/n”)

leer (aux)

mientras ( (aux<>‘s’) Y (aux<>’n’) ) hacer

escribir (“Opción incorrecta. ¿Desea modificar la aerolínea? s/n”)

leer (aux)

fin mientras

**Fin Método**

**Fin Clase**

Persona= **Clas**e de

cedula: entero

nombre,apellido,telefono: cadena

fecha\_nac: fecha

**Método** llenar()

**Inicio**

escribir(“Cédula”)

leer (cedula)

escribir(“Nombre”)

leer (nombre)

escribir(“Apellido”)

leer (apellido)

escribir (“Fecha de Nacimiento”)

fecha\_nac←nuevo(Fecha)

fecha\_nac.llenar()

escribir(“Teléfono”)

leer (telefono)

**Fin Método**

**Fin Clase**

Vuelo=**Clase** de

id : cadena

fecha\_vuelo:fecha

hora\_salida: hora

hora\_llegada: hora

línea: cadena

pilotos: Arreglo [10] de Piloto

azaf: Arreglo[10] de Azafat

cli: Arreglo [1000] de Persona

id\_avion: cadena

**Método** llenar()

**Inicio**

escribir (“Ingrese id”)

leer (id)

fecha\_vuelo←nuevo (Fecha)

fecha\_vuelo.llenar()

hora\_salida←nuevo(Hora)

hora\_salida.llenar()

hora\_llegada←nuevo(Hora)

hora\_llegada.llenar()

mientras (hora\_llegada.comparar(hora\_salida)=falso) hacer

escribir(“La hora de llegada debe ser mayor que la de salida ”)

hora\_llegada.llenar()

fin mientras

escribir (“Nombre de la aerolínea”)

leer (linea)

**Fin Método**

**Método** encontrar\_linea (arrlin[]:Linea): booleano

**Variables**

i:entero

**Inicio**

para i (1 hasta 30) hacer

si (arrlin[i]<>NULO) entonces

si (arrlin[i].nombre=linea) entonces

encontrar\_linea←verdadero

fin si

fin si

fin para

encontrar\_linea←falso

**Fin Método**

**Fin Clase**

**Tipos**

arravion= Arreglo [30][15] de Avion

arrlin= Arreglo [30] de Linea

**Variables**

n.lineas,n\_aviones,i,j: entero

**Inicio**

escribir (“Ingrese la cantidad de aerolíneas que desee registrar”)

leer(n\_lineas)

mientras ((n\_lineas<=0) O (n\_lineas>30)) hacer

escribir (“La cantidad de aerolíneas debe estar entre 1 y 30”)

escribir (“Ingrese nuevamente el valor”)

leer (n\_lineas)

fin mientras

para i (1 hasta n\_lineas) hacer

escribir (“aerolínea nº ”,i)

arrlin[i]←nuevo(Línea)

arrlin[i].llenar()

si (i>1) entonces

repetdo←verdadero

mientras (repetdo=verdadero)

repetdo←falso

para j (1 hasta i-1) hacer

si (arrlin[j].nombre= arrlin[i].nombre) entonces

repetido←verdadero

fin si

fin para

si (repetido=verdadero) entonces

escribir (“Nombre repetido. Ingréselo de nuevo”)

leer (arrlin[i].nombre)

fin si

fin mientras

fin si

fin para

escribir (“Ingrese la cantidad de aviones por aerolínea”)

leer(n\_aviones)

mientras ((n\_ aviones <=0) O (n\_ aviones >15)) hacer

escribir (“La cantidad de avioines debe estar entre 1 y 15”)

escribir (“Ingrese nuevamente el valor”)

leer (n\_ aviones)

fin mientras

para i (1 hasta n\_lineas) hacer

para j (1 hasta n\_aviones) hacer

arravion[i][j]←nuevo(Avion)

escribir (“Avión nº ”,j,” de la aerolínea ”,arrlin[i].nombre)

arravion[i][j].llenar()

si (i\*j<n\_lineas\*n\_aviones) hacer

repetido←verdadero

mientras (repetido=verdadero)

repetido←falso

para k (1 hasta i) hacer

para l (1 hasta j-1) hacer

si (arravion[k][l].id= arravion [i][j].id)

entonces

repetido←verdadero

fin si

fin para

fin para

si (repetido=verdadero) entonces

escribir (“Id repetido. Ingréselo de nuevo”)

leer (arravion [i][j].id)

fin si

fin mientras

fin si

fin para

fin para

escribir (“Ingrese la cantidad de pilotos que desee registrar por aerolinea”)

leer(n\_pilotos)

mientras ((n\_ pilotos <=0) O (n\_ pilotos >10) hacer

escribir (“La cantidad de pilotos debe estar entre 1 y 10”)

escribir (“Ingrese nuevamente el valor”)

leer (n\_ pilotos)

fin mientras

para i (1 hasta n\_lineas) hacer

para j (1 hasta n\_pilotos) hacer

arrpiloto[i][j]←nuevo(Piloto)

escribir (“Piloto nº ”,j,” de la aerolínea ”,arrlin[i].nombre)

arrpiloto[i][j].llenar()

si (i\*j<n\_lineas\*n\_pilotos) hacer

repetido←verdadero

mientras (repetido=verdadero)

repetido←falso

para k (1 hasta i) hacer

para l (1 hasta j-1) hacer

si (arrpiloto [k][l].cedula= arrpiloto[i][j].cedula)

entonces

repetido←verdadero

fin si

fin para

fin para

si (repetido=verdadero) entonces

escribir (“Cédula repetida. Ingrésela de

nuevo”)

leer (arrpiloto [i][j].cedula)

fin si

fin mientras

fin si

fin para

fin para

escribir (“Ingrese la cantidad de azafatas/os que desee registrar por aerolinea”)

leer(n\_azafat)

mientras ((n\_azafat <=0) O (n\_azafat >10) hacer

escribir (“La cantidad de pilotos debe estar entre 1 y 10”)

escribir (“Ingrese nuevamente el valor”)

leer (n\_azafat)

fin mientras

para i (1 hasta n\_lineas) hacer

para j (1 hasta n\_azafat) hacer

arrpiloto[i][j]←nuevo(Piloto)

escribir (“Azafata/o nº ”,j,” de la aerolínea ”,arrlin[i].nombre)

arrazafat[i][j].llenar()

si (i\*j<n\_lineas\*n\_azafat) hacer

repetido←verdadero

mientras (repetido=verdadero)

repetido←falso

para k (1 hasta i) hacer

para l (1 hasta j-1) hacer

si (arrazafat [k][l].cedula= arrazafat

[i][j].cedula) entonces

repetido←verdadero

fin si

fin para

fin para

si (repetido=verdadero) entonces

escribir (“Cedula repetida. Ingrésela de nuevo”)

leer (arrazafat [i][j].id)

fin si

fin mientras

fin si

fin para

fin para

respuesta←’n’

i←1

mientras (respuesta=’n’) hacer

arrvuelo[i]←nuevo(Vuelo)

arrvuelo[i].llenar()

mientras (arrvuelo[i ].encontrar\_linea(arrlin)=falso) hacer

escribir(“línea no encontrada, ingrésela de nuevo”)

leer arrvuelo[i].linea

fin mientras

escribir (“Id del avión”)

leer (arrvuelo[i].id\_avion)

repetido←falso

mientras (repetido=falso) hacer

para j(1 hasta n\_linea) hacer

para k(1 hasta n\_avion) hacer

si ( (arravion[j][k].id =arrvuelo[i].id\_avion)

Y(arrvuelo[i].linea=arravion[j][k].linea) )entonces

repetido←verdadero

fin si

fin para

fin para

si (repetido=falso) entonces

escribir (”id no encontrado. Ingréselo nuevamente”)

leer (arrvuelo[i].id\_avion)

fin si

fin mientras

escribir (“cantidad de azafatas/os en el vuelo”)

leer n\_azafat\_vuelo

mientras (n\_azafat\_vuelo<=0) O (n\_azafat\_vuelo>10) hacer

escribir (“La cantidad debe oscilar entre 1 y 10. Ingrésela de nuevo”)

leer (n\_azafat\_vuelo)

fin mientras

para j (1 hasta n\_azafat\_vuelo) hacer

arrvuelo[j].azaf[j]←nuevo(Azafat)

escribir (“cédula de la azafata/o”)

leer (arrvuelo[i].azaf[j].cedula)

repetido←falso

mientras (repetido=falso) hacer

repetido←verdadero

parak(1 hasta n\_linea) hacer

para l(1 hasta n\_azafat) hacer

si( (arrvuelo[i].azaf[j].cedula=

arrazafat[k][l].cedula)) Y

(arrvuelo[i].linea=arrazafat[k][l].linea)

)entonces

repetido←verdadero

fin si

fin para

fin para

si (repetido=falso) entonces

escribir (”Cedula no encontrada. Ingrésela

nuevamente”)

leer (arrvuelo[i].azaf[j].cedula)

fin si

fin mientras

fin para

escribir (“cantidad de pilotos en el vuelo”)

leer n\_pilotos\_vuelo

mientras (n\_pilotos\_vuelo<=0) O (n\_ pilotos \_vuelo>10) hacer

escribir (“La cantidad debe oscilar entre 1 y 10. Ingrésela de nuevo”)

leer (n\_ pilotos \_vuelo)

fin mientras

para j (1 hasta n\_ pilotos \_vuelo) hacer

arrvuelo[j].pilotos[j]←nuevo(Piloto)

escribir (“cédula de la azafata/o”)

leer (arrvuelo[i].pilotos[j].cedula)

repetido←falso

mientras (repetido=falso) hacer

repetido←verdadero

parak(1 hasta n\_linea) hacer

para l(1 hasta n\_pilotos) hacer

si( (arrvuelo[i].pilotos [j].cedula=

arrpiloto[k][l].cedula)) Y

(arrvuelo[i].linea=arrpiloto[k][l].linea)

)entonces

repetido←verdadero

fin si

fin para

fin para

si (repetido=falso) entonces

escribir (”Cedula no encontrada. Ingrésela

nuevamente”)

leer (arrvuelo[i].piloto[j].cedula)

fin si

fin mientras

fin para

escribir (“cantidad de personas en el vuelo”)

leer n\_personas\_vuelo

mientras (n\_ personas \_vuelo<=0) O (n\_ personas \_vuelo>1000) hacer

escribir (“La cantidad debe oscilar entre 1 y 1000. Ingrésela de nuevo”)

leer (n\_ personas \_vuelo)

fin mientras

para j (1 hasta \_ personas \_vuelo) hacer

escribir (“cliente nº ”,i)

arrvuelo[i].cli[j]←nuevo(Persona)

arrvuelo[i].cli[j].llenar()

si (j>1) entonces

repetido←verdadero

mientras (repetdo=verdadero)

repetido←falso

para l (1 hasta j-1) hacer

si (arrvuelo[i].cli[j].cedula=

arrvuelo[i].cli[l].cedula) entonces

repetido←verdadero

fin si

fin para

si (repetido=verdadero) entonces

escribir (“Cédula repetida. Ingrésela de nuevo”)

leer (arrvuelo[i].cli[j].cedula)

fin si

fin mientras

fin si

fin para

fin mientras

**Fin Algoritmo**