

Universidad ORT Uruguay – Facultad de Ingeniería

Obligatorio 1 – Programación 2

Joaquin Peluffo



Profesor: Alejandro Morales

Fecha: 12/05/2025

Contenido

Universidad ORT Uruguay – Facultad de Ingeniería	1
Índice	2
Tabla de precarga:.....	3
//creacion de admin y cliente.....	3
//CREA OCACIONALES	3
//CREA PREMIUM	3
//creacion de aeropuerto	3
//creacion de ruta.....	4
//creacion de avion.....	5
//creacion de vuelo.....	6
//creacion de pasaje.....	7
Evidencia IA usada:	10
Diagrama UML	12
CODIGO.....	14
//PROGRAM	14
//METODOS	16
//Menus.....	28
//CREAR EJEMPLOS.....	32
//SISTEMA.....	39
//Vuelo.....	52
//Usuario	55
//Ruta.....	58
//Premium	61
//Pasaje.....	62
//Ocacional	63
//Frecuencia Vuelo	65
//Equipaje	66
//CLIENTE.....	66
//AVION.....	68
//AEROPUERTO	72
//ADMINISTRADOR	75

Tabla de precarga:

//creacion de admin y cliente

```
system.createAdmin("joaco@zionbancorp.com", "joaco", "pass", "joacoApodo");
```

```
system.createAdmin("isma@microsoft.com", "isma", "pass", "ismaApodo");
```

//CREA OCACIONALES

```
Cliente pasajero1 = system.createClienteOccasional("ezequieltitan@hotmail.com",  
"eze", "pass", 46689505, "Ezequiel", "Uruguayo");
```

```
Cliente pasajero2 = system.createClienteOccasional("sofia@mail.com", "sofi", "1234",  
45321231, "Sofia", "Uruguay");
```

```
Cliente pasajero3 = system.createClienteOccasional("marco@mail.com", "marco123",  
"abcd", 48976543, "Marco", "Chileno");
```

```
Cliente pasajero4 = system.createClienteOccasional("lucia@mail.com", "lucy", "qwerty",  
45511222, "Lucia", "Paraguay");
```

```
Cliente pasajero5 = system.createClienteOccasional("diego@mail.com", "dieguito",  
"zxcv", 44112233, "Diego", "Boliviano");
```

//CREA PREMIUM

```
Cliente pasajero6 = system.createClientePremium("estimail@premium.com", "esti",  
"pass", 12332126, "Esteban", "Argentino");
```

```
Cliente pasajero7 = system.createClientePremium("ana@mail.com", "anita",  
"anapass", 99887766, "Ana", "Uruguay");
```

```
Cliente pasajero8 = system.createClientePremium("jose@mail.com", "jose", "josito",  
11223344, "Jose", "Peruano");
```

```
Cliente pasajero9 = system.createClientePremium("carla@mail.com", "carli", "car123",  
55667788, "Carla", "Venezolana");
```

```
Cliente pasajero10 = system.createClientePremium("nicolas@mail.com", "nico",  
"nicopass", 77889900, "Nicolas", "Colombiano");
```

//creacion de aeropuerto

```
Aeropuerto aeropuertoUYU = system.createAeropuerto("UYU", "Montevideo", 80, 17);
```

```
Aeropuerto aeropuertoARG = system.createAeropuerto("ARG", "Buenos Aires", 120,  
12);
```

```
Aeropuerto aeropuertoUSA = system.createAeropuerto("USA", "New York", 77, 24);  
Aeropuerto aeropuertoNIG = system.createAeropuerto("NIG", "Lagos", 33, 6);  
Aeropuerto aeropuertoBRA = system.createAeropuerto("BRA", "Rio de Janeiro", 95,  
15);  
Aeropuerto aeropuertoMEX = system.createAeropuerto("MEX", "Ciudad de México",  
85, 13);  
Aeropuerto aeropuertoCAN = system.createAeropuerto("CAN", "Toronto", 105, 18);  
Aeropuerto aeropuertoFRA = system.createAeropuerto("FRA", "Frankfurt", 150, 22);  
Aeropuerto aeropuertoESP = system.createAeropuerto("ESP", "Madrid", 110, 14);  
Aeropuerto aeropuertoITA = system.createAeropuerto("ITA", "Roma", 100, 16);  
Aeropuerto aeropuertoCHL = system.createAeropuerto("CHL", "Santiago", 90, 11);  
Aeropuerto aeropuertoPER = system.createAeropuerto("PER", "Lima", 88, 10);  
Aeropuerto aeropuertoCOL = system.createAeropuerto("COL", "Bogotá", 87, 9);  
Aeropuerto aeropuertoVEN = system.createAeropuerto("VEN", "Caracas", 70, 8);  
Aeropuerto aeropuertoURU = system.createAeropuerto("URU", "Punta del Este", 60,  
7);  
Aeropuerto aeropuertoUK = system.createAeropuerto("UK", "Londres", 140, 25);  
Aeropuerto aeropuertoJPN = system.createAeropuerto("JPN", "Tokio", 160, 30);  
Aeropuerto aeropuertoCHN = system.createAeropuerto("CHN", "Pekín", 155, 28);  
Aeropuerto aeropuertoAUS = system.createAeropuerto("AUS", "Sídney", 145, 20);  
Aeropuerto aeropuertoIND = system.createAeropuerto("IND", "Nueva Delhi", 130, 19);
```

//creacion de ruta

```
Ruta rutaUyuArg = system.createRuta(aeropuertoUYU, aeropuertoARG, 1000);  
Ruta rutaUyuBra = system.createRuta(aeropuertoUYU, aeropuertoBRA, 1800);  
Ruta rutaUyuChl = system.createRuta(aeropuertoUYU, aeropuertoCHL, 1600);  
Ruta rutaArgUsa = system.createRuta(aeropuertoARG, aeropuertoUSA, 8500);  
Ruta rutaArgFra = system.createRuta(aeropuertoARG, aeropuertoFRA, 11000);  
Ruta rutaArgEsp = system.createRuta(aeropuertoARG, aeropuertoESP, 10200);  
Ruta rutaBraMex = system.createRuta(aeropuertoBRA, aeropuertoMEX, 7400);
```

```
Ruta rutaBraNig = system.createRuta(aeropuertoBRA, aeropuertoNIG, 6800);
Ruta rutaChlPer = system.createRuta(aeropuertoCHL, aeropuertoPER, 2500);
Ruta rutaChlCol = system.createRuta(aeropuertoCHL, aeropuertoCOL, 2700);
Ruta rutaMexCan = system.createRuta(aeropuertoMEX, aeropuertoCAN, 3600);
Ruta rutaCanUsa = system.createRuta(aeropuertoCAN, aeropuertoUSA, 800);
Ruta rutaUsaFra = system.createRuta(aeropuertoUSA, aeropuertoFRA, 5900);
Ruta rutaUsaEsp = system.createRuta(aeropuertoUSA, aeropuertoESP, 6100);
Ruta rutaUsalta = system.createRuta(aeropuertoUSA, aeropuertoITA, 7000);
Ruta rutaFraUk = system.createRuta(aeropuertoFRA, aeropuertoUK, 900);
Ruta rutaEspUk = system.createRuta(aeropuertoESP, aeropuertoUK, 1300);
Ruta rutaUkJpn = system.createRuta(aeropuertoUK, aeropuertoJPN, 9700);
Ruta rutaUkChn = system.createRuta(aeropuertoUK, aeropuertoCHN, 8900);
Ruta rutaJpnAus = system.createRuta(aeropuertoJPN, aeropuertoAUS, 7800);
Ruta rutaChnInd = system.createRuta(aeropuertoCHN, aeropuertoIND, 4800);
Ruta rutaIndAus = system.createRuta(aeropuertoIND, aeropuertoAUS, 10100);
Ruta rutaIndNig = system.createRuta(aeropuertoIND, aeropuertoNIG, 7500);
Ruta rutaNigVen = system.createRuta(aeropuertoNIG, aeropuertoVEN, 9200);
Ruta rutaVenCol = system.createRuta(aeropuertoVEN, aeropuertoCOL, 1600);
Ruta rutaColPer = system.createRuta(aeropuertoCOL, aeropuertoPER, 1800);
Ruta rutaPerEsp = system.createRuta(aeropuertoPER, aeropuertoESP, 9600);
Ruta rutaCanFra = system.createRuta(aeropuertoCAN, aeropuertoFRA, 6000);
Ruta rutaUyuUk = system.createRuta(aeropuertoUYU, aeropuertoUK, 11400);
Ruta rutaEspJpn = system.createRuta(aeropuertoESP, aeropuertoJPN, 10800);
```

//creacion de avion

```
Avion primerAvion = system.createAvion("Nissan", "modeloJapones", 200, 50, 5);
Avion segundoAvion = system.createAvion("Boeing", "737 Max", 180, 6500, 8.5m);
Avion tercerAvion = system.createAvion("Airbus", "A320neo", 190, 6100, 8.0m);
```

```
Avion cuartoAvion = system.createAvion("Embraer", "E195-E2", 132, 4800, 6.5m);
```

//creacion de vuelo

```
Vuelo vuelo1 = system.createVuelo("AA334", rutaUyuArg, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes, FrecuenciaVuelo.Martes });
```

```
Vuelo vuelo2 = system.createVuelo("BB101", rutaUyuBra, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles, FrecuenciaVuelo.Viernes });
```

```
Vuelo vuelo3 = system.createVuelo("CC202", rutaUyuChl, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Sabado });
```

```
Vuelo vuelo4 = system.createVuelo("DD303", rutaArgUsa, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });
```

```
Vuelo vuelo5 = system.createVuelo("EE404", rutaArgFra, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo6 = system.createVuelo("FF505", rutaArgEsp, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Viernes });
```

```
Vuelo vuelo7 = system.createVuelo("GG606", rutaBraMex, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });
```

```
Vuelo vuelo8 = system.createVuelo("HH707", rutaBraNig, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles, FrecuenciaVuelo.Sabado });
```

```
Vuelo vuelo9 = system.createVuelo("II808", rutaChlPer, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });
```

```
Vuelo vuelo10 = system.createVuelo("JJ909", rutaChlCol, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });
```

```
Vuelo vuelo11 = system.createVuelo("KK111", rutaMexCan, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles, FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo12 = system.createVuelo("LL222", rutaCanUsa, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Viernes });
```

```
Vuelo vuelo13 = system.createVuelo("MM333", rutaUsaFra, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo14 = system.createVuelo("NN444", rutaUsaEsp, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Sabado });
```

```
Vuelo vuelo15 = system.createVuelo("OO555", rutaUsalta, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });
```

```
Vuelo vuelo16 = system.createVuelo("PP666", rutaFraUk, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });
```

```
Vuelo vuelo17 = system.createVuelo("QQ777", rutaEspUk, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo18 = system.createVuelo("RR888", rutaUkJpn, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Viernes, FrecuenciaVuelo.Sabado });
```

```
Vuelo vuelo19 = system.createVuelo("SS999", rutaUkChn, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles });
```

```
Vuelo vuelo20 = system.createVuelo("TT000", rutaJpnAus, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes });
```

```
Vuelo vuelo21 = system.createVuelo("UU121", rutaChnInd, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });
```

```
Vuelo vuelo22 = system.createVuelo("VV232", rutaIndAus, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });
```

```
Vuelo vuelo23 = system.createVuelo("WW343", rutaIndNig, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles });
```

```
Vuelo vuelo24 = system.createVuelo("XX454", rutaNigVen, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo25 = system.createVuelo("YY565", rutaVenCol, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Sabado });
```

```
Vuelo vuelo26 = system.createVuelo("ZZ676", rutaColPer, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes, FrecuenciaVuelo.Miercoles });
```

```
Vuelo vuelo27 = system.createVuelo("AB787", rutaPerEsp, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Viernes });
```

```
Vuelo vuelo28 = system.createVuelo("BC898", rutaCanFra, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });
```

```
Vuelo vuelo29 = system.createVuelo("CD909", rutaUyuUk, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo30 = system.createVuelo("DE010", rutaEspJpn, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Sabado });
```

```
//creacion de pasaje
```

```
Pasaje pasaje1 = system.createPasaje(vuelo1, new DateTime(2003, 09, 10), pasajero1,  
Equipaje.BODEGA);
```

Pasaje pasaje2 = system.createPasaje(vuelo2, new DateTime(2004, 10, 12), pasajero2, Equipaje.LIGHT);

Pasaje pasaje3 = system.createPasaje(vuelo3, new DateTime(2005, 11, 14), pasajero3, Equipaje.CABINA);

Pasaje pasaje4 = system.createPasaje(vuelo4, new DateTime(2006, 12, 16), pasajero4, Equipaje.LIGHT);

Pasaje pasaje5 = system.createPasaje(vuelo5, new DateTime(2007, 01, 18), pasajero5, Equipaje.BODEGA);

Pasaje pasaje6 = system.createPasaje(vuelo6, new DateTime(2008, 02, 20), pasajero6, Equipaje.CABINA);

Pasaje pasaje7 = system.createPasaje(vuelo7, new DateTime(2009, 03, 22), pasajero7, Equipaje.BODEGA);

Pasaje pasaje8 = system.createPasaje(vuelo8, new DateTime(2010, 04, 24), pasajero8, Equipaje.LIGHT);

Pasaje pasaje9 = system.createPasaje(vuelo9, new DateTime(2011, 05, 26), pasajero9, Equipaje.CABINA);

Pasaje pasaje10 = system.createPasaje(vuelo10, new DateTime(2012, 06, 28), pasajero10, Equipaje.BODEGA);

Pasaje pasaje11 = system.createPasaje(vuelo11, new DateTime(2013, 07, 30), pasajero1, Equipaje.LIGHT);

Pasaje pasaje12 = system.createPasaje(vuelo12, new DateTime(2014, 08, 02), pasajero2, Equipaje.CABINA);

Pasaje pasaje13 = system.createPasaje(vuelo13, new DateTime(2015, 09, 04), pasajero3, Equipaje.BODEGA);

Pasaje pasaje14 = system.createPasaje(vuelo14, new DateTime(2016, 10, 06), pasajero4, Equipaje.LIGHT);

Pasaje pasaje15 = system.createPasaje(vuelo15, new DateTime(2017, 11, 08), pasajero5, Equipaje.CABINA);

Pasaje pasaje16 = system.createPasaje(vuelo16, new DateTime(2018, 12, 10), pasajero6, Equipaje.BODEGA);

Pasaje pasaje17 = system.createPasaje(vuelo17, new DateTime(2019, 01, 12), pasajero7, Equipaje.LIGHT);

Pasaje pasaje18 = system.createPasaje(vuelo18, new DateTime(2020, 02, 14),
pasajero8, Equipaje.CABINA);

Pasaje pasaje19 = system.createPasaje(vuelo19, new DateTime(2021, 03, 16),
pasajero9, Equipaje.BODEGA);

Pasaje pasaje20 = system.createPasaje(vuelo20, new DateTime(2022, 04, 18),
pasajero10, Equipaje.LIGHT);

Pasaje pasaje21 = system.createPasaje(vuelo21, new DateTime(2023, 05, 20),
pasajero1, Equipaje.CABINA);

Pasaje pasaje22 = system.createPasaje(vuelo22, new DateTime(2024, 06, 22),
pasajero2, Equipaje.BODEGA);

Pasaje pasaje23 = system.createPasaje(vuelo23, new DateTime(2025, 07, 24),
pasajero3, Equipaje.LIGHT);

Pasaje pasaje24 = system.createPasaje(vuelo24, new DateTime(2026, 08, 26),
pasajero4, Equipaje.CABINA);

Pasaje pasaje25 = system.createPasaje(vuelo25, new DateTime(2027, 09, 28),
pasajero5, Equipaje.BODEGA);

Evidencia IA usada:

Memoria guardada llena. O
Cree 5 de cada uno, ocasional y premium:
//CREA OCACIONALES
system.createClienteOcasional("ezequiel@hotmail.com",
"eze", "pass", 46689505, "ezequiel", "Uruguayo");
//CREA PREMIUM
system.createClientePremium("estimal@premium.com", "esti",
"pass", 1233212, "Esteban", "Argentino");

Aquí tienes el código para crear 5 clientes Ocasionales y 5 clientes Premium con datos variados:

csharp

Copiar

Editar

```
// CREA OCACIONALES
system.createClienteOcasional("ezequiel@hotmail.com", "eze", "pass", 46689505, "Ezequiel", "Uruguayo");
system.createClienteOcasional("ana.ocasional@gmail.com", "ana123", "anaPass", 41022567, "Ana", "Uruguayo");
system.createClienteOcasional("lucas.ocasional@gmail.com", "lucas", "lucaspass", 42111222, "Lucas", "Uruguayo");
system.createClienteOcasional("carla.q@hotmail.com", "carlaq", "1234", 47895432, "Carla", "Peruano");
system.createClienteOcasional("ocasional@correo.com", "users", "clave", 49873120, "Pedro", "Boliviano");

// CREA PREMIUM
system.createClientePremium("estimal@premium.com", "esti", "pass", 1233212, "Esteban", "Argentino");
system.createClientePremium("premium.ana@gmail.com", "anap", "prepass", 41332255, "Ana P", "Uruguayo");
system.createClientePremium("jorge_premium@hotmail.com", "jorgep", "123456", 45612345, "Jorge", "Uruguayo");
system.createClientePremium("maria_prem@correo.com", "marlap", "premi123", 41234567, "Maria", "Paraguayo");
system.createClientePremium("cliente@premium.com", "cli10", "clavep", 46789988, "Luis", "Ecuatoriano");
```

Ahora quiero 30 rutas en las que estas son las props:Aeropuerto
aeropuertoSalida, Aeropuerto aeropuertoEntrada, decimal distancia
Con este ejemplo:
Ruta rutaYyuArg = system.createRuta(aeropuertoUYU,
aeropuertoARG, 1000);

Perfecto, aquí tienes 30 rutas creadas con tu estructura `system.createRuta(...)` y usando variables como en tu ejemplo:

csharp

Copiar

Editar

```
Ruta rutaYyuArg = system.createRuta(aeropuertoUYU, aeropuertoARG, 1000);
Ruta rutaYyuBra = system.createRuta(aeropuertoUYU, aeropuertoBRA, 1800);
Ruta rutaYyuChl = system.createRuta(aeropuertoUYU, aeropuertoCHL, 1600);
Ruta rutaArgUsa = system.createRuta(aeropuertoARG, aeropuertoUSA, 8500);
Ruta rutaArgFra = system.createRuta(aeropuertoARG, aeropuertoFRA, 11000);
Ruta rutaArgEsp = system.createRuta(aeropuertoARG, aeropuertoESP, 10200);
Ruta rutaBraMex = system.createRuta(aeropuertoBRA, aeropuertoMEX, 7400);
Ruta rutaBraNig = system.createRuta(aeropuertoBRA, aeropuertoNIG, 6000);
Ruta rutaChlPer = system.createRuta(aeropuertoCHL, aeropuertoPER, 2500);
Ruta rutaChlCol = system.createRuta(aeropuertoCHL, aeropuertoCOL, 2700);
Ruta rutaMexCan = system.createRuta(aeropuertoMEX, aeropuertoCAN, 3600);
Ruta rutaCanUsa = system.createRuta(aeropuertoCAN, aeropuertoUSA, 800);
Ruta rutaUsaFra = system.createRuta(aeropuertoUSA, aeropuertoFRA, 5900);
Ruta rutaUsaEsp = system.createRuta(aeropuertoUSA, aeropuertoESP, 6100);
Ruta rutaUsaIta = system.createRuta(aeropuertoUSA, aeropuertoITA, 7000);
Ruta rutaFraUk = system.createRuta(aeropuertoFRA, aeropuertoUK, 900);
Ruta rutaEspUk = system.createRuta(aeropuertoESP, aeropuertoUK, 1300);
Ruta rutaUkJpn = system.createRuta(aeropuertoUK, aeropuertoJPN, 9700);
Ruta rutaUkChn = system.createRuta(aeropuertoUK, aeropuertoCHN, 8900);
```

Perfecto, aquí tienes los 20 aeropuertos creados con la estructura que usas en tu ejemplo, guardando cada uno en una variable:

csharp

Copiar

Editar

```
Aeropuerto aeropuertoUYU = system.createAeropuerto("UYU", "Montevideo", 80, 17);
Aeropuerto aeropuertoARG = system.createAeropuerto("ARG", "Buenos Aires", 120, 12);
Aeropuerto aeropuertoUSA = system.createAeropuerto("USA", "New York", 77, 24);
Aeropuerto aeropuertoNIG = system.createAeropuerto("NIG", "Lagos", 33, 6);
Aeropuerto aeropuertoBRA = system.createAeropuerto("BRA", "Rio de Janeiro", 95, 15);
Aeropuerto aeropuertoMEX = system.createAeropuerto("MEX", "Ciudad de México", 85, 13);
Aeropuerto aeropuertoCAN = system.createAeropuerto("CAN", "Toronto", 105, 18);
Aeropuerto aeropuertoFRA = system.createAeropuerto("FRA", "Frankfurt", 150, 22);
Aeropuerto aeropuertoESP = system.createAeropuerto("ESP", "Madrid", 110, 14);
Aeropuerto aeropuertoITA = system.createAeropuerto("ITA", "Roma", 100, 16);
Aeropuerto aeropuertoCHL = system.createAeropuerto("CHL", "Santiago", 90, 11);
Aeropuerto aeropuertoPER = system.createAeropuerto("PER", "Lima", 88, 10);
Aeropuerto aeropuertoCOL = system.createAeropuerto("COL", "Bogotá", 87, 9);
Aeropuerto aeropuertoVEN = system.createAeropuerto("VEN", "Caracas", 70, 0);
Aeropuerto aeropuertoURU = system.createAeropuerto("URU", "Punta del Este", 60, 7);
Aeropuerto aeropuertoUK = system.createAeropuerto("UK", "Londres", 140, 25);
Aeropuerto aeropuertoJPN = system.createAeropuerto("JPN", "Tokio", 160, 30);
Aeropuerto aeropuertoCHN = system.createAeropuerto("CHN", "Pekín", 155, 28);
Aeropuerto aeropuertoAUS = system.createAeropuerto("AUS", "Sídney", 145, 20);
Aeropuerto aeropuertoIND = system.createAeropuerto("IND", "Nueva Delhi", 130, 19);
```

Ahora 4 aviones como este ejemplo:

```
Avion primerAvion = system.createAvion("Nissan",
"modeloJapones", 200, 50, 5);
Estas son las props:
string fabricante, string modelo, int cantAsientos, decimal alcanceKm,
decimal costoKm
```

Perfecto, aquí tienes 4 aviones creados siguiendo el formato que mencionaste:

```
csharp Copiar Editar

Avion primerAvion = system.createAvion("Nissan", "modeloJapones", 200, 50, 5);
Avion segundoAvion = system.createAvion("Boeing", "737 Max", 180, 6500, 8.5m);
Avion tercerAvion = system.createAvion("Airbus", "A320neo", 190, 6100, 8.6m);
Avion cuartoAvion = system.createAvion("Embraer", "E195-E2", 132, 4800, 6.5m);
```

Perfecto, aquí tienes los 25 pasajes con asignación a variables siguiendo tu ejemplo:

```
csharp Copiar Editar

Pasaje pasaje1 = system.createPasaje(vuelo1, new DateTime(2003, 09, 10), pasajero1, Equipaje.BODEGA);
Pasaje pasaje2 = system.createPasaje(vuelo2, new DateTime(2004, 10, 12), pasajero2, Equipaje.LIGHT);
Pasaje pasaje3 = system.createPasaje(vuelo3, new DateTime(2005, 11, 14), pasajero3, Equipaje.CABINA);
Pasaje pasaje4 = system.createPasaje(vuelo4, new DateTime(2006, 12, 16), pasajero4, Equipaje.LIGHT);
Pasaje pasaje5 = system.createPasaje(vuelo5, new DateTime(2007, 01, 18), pasajero5, Equipaje.BODEGA);
Pasaje pasaje6 = system.createPasaje(vuelo6, new DateTime(2008, 02, 20), pasajero6, Equipaje.CABINA);
Pasaje pasaje7 = system.createPasaje(vuelo7, new DateTime(2009, 03, 22), pasajero7, Equipaje.BODEGA);
Pasaje pasaje8 = system.createPasaje(vuelo8, new DateTime(2010, 04, 24), pasajero8, Equipaje.LIGHT);
Pasaje pasaje9 = system.createPasaje(vuelo9, new DateTime(2011, 05, 26), pasajero9, Equipaje.CABINA);
Pasaje pasaje10 = system.createPasaje(vuelo10, new DateTime(2012, 06, 28), pasajero10, Equipaje.BODEGA);
Pasaje pasaje11 = system.createPasaje(vuelo11, new DateTime(2013, 07, 30), pasajero1, Equipaje.LIGHT);
Pasaje pasaje12 = system.createPasaje(vuelo12, new DateTime(2014, 08, 02), pasajero2, Equipaje.CABINA);
Pasaje pasaje13 = system.createPasaje(vuelo13, new DateTime(2015, 09, 04), pasajero3, Equipaje.BODEGA);
Pasaje pasaje14 = system.createPasaje(vuelo14, new DateTime(2016, 10, 06), pasajero4, Equipaje.LIGHT);
Pasaje pasaje15 = system.createPasaje(vuelo15, new DateTime(2017, 11, 08), pasajero5, Equipaje.CABINA);
Pasaje pasaje16 = system.createPasaje(vuelo16, new DateTime(2018, 12, 10), pasajero6, Equipaje.BODEGA);
Pasaje pasaje17 = system.createPasaje(vuelo17, new DateTime(2019, 01, 12), pasajero7, Equipaje.LIGHT);
```

Ahora 30 vuelos como este son las props:

```
string numeroVuelo, Ruta ruta, Avion avion, List<FrecuenciaVuelo> frecuencia
Este es un ejemplo
Vuelo vuelo1 = system.createVuelo("AA334", rutaUyuArg,
primerAvion, new List<FrecuenciaVuelo>
{FrecuenciaVuelo.Lunes,FrecuenciaVuelo.Martes});
```

Aquí tienes 30 vuelos creados con diferentes rutas, aviones y frecuencias variadas, siguiendo tu ejemplo:

```
csharp Copiar Editar

Vuelo vuelo1 = system.createVuelo("AA334", rutaUyuArg, primerAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Lunes, FrecuenciaVuelo.Martes});
Vuelo vuelo2 = system.createVuelo("BB101", rutaUyuBra, segundoAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Martes, FrecuenciaVuelo.Miércoles});
Vuelo vuelo3 = system.createVuelo("CC202", rutaUyuChl, tercerAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Miércoles, FrecuenciaVuelo.Jueves});
Vuelo vuelo4 = system.createVuelo("DD303", rutaArgUsa, cuartoAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Jueves, FrecuenciaVuelo.Viernes});
Vuelo vuelo5 = system.createVuelo("EE404", rutaArgFra, segundoAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Viernes, FrecuenciaVuelo.Sábado});
Vuelo vuelo6 = system.createVuelo("FF505", rutaArgEsp, tercerAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Sábado, FrecuenciaVuelo.Domingo});
Vuelo vuelo7 = system.createVuelo("GG606", rutaBraMex, cuartoAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Domingo, FrecuenciaVuelo.Lunes});
Vuelo vuelo8 = system.createVuelo("HH707", rutaBraNig, primerAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Lunes, FrecuenciaVuelo.Martes});
Vuelo vuelo9 = system.createVuelo("II808", rutaChlPer, segundoAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Martes, FrecuenciaVuelo.Miércoles});
Vuelo vuelo10 = system.createVuelo("JJ909", rutaChlCol, tercerAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Miércoles, FrecuenciaVuelo.Jueves});
Vuelo vuelo11 = system.createVuelo("KK111", rutaMexCan, cuartoAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Jueves, FrecuenciaVuelo.Viernes});
Vuelo vuelo12 = system.createVuelo("LL222", rutaCanUsa, primerAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Viernes, FrecuenciaVuelo.Sábado});
Vuelo vuelo13 = system.createVuelo("MW333", rutaUsaFra, segundoAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Sábado, FrecuenciaVuelo.Domingo});
Vuelo vuelo14 = system.createVuelo("WW444", rutaUsaEsp, tercerAvion, new List<FrecuenciaVuelo> {FrecuenciaVuelo.Domingo, FrecuenciaVuelo.Lunes});
```

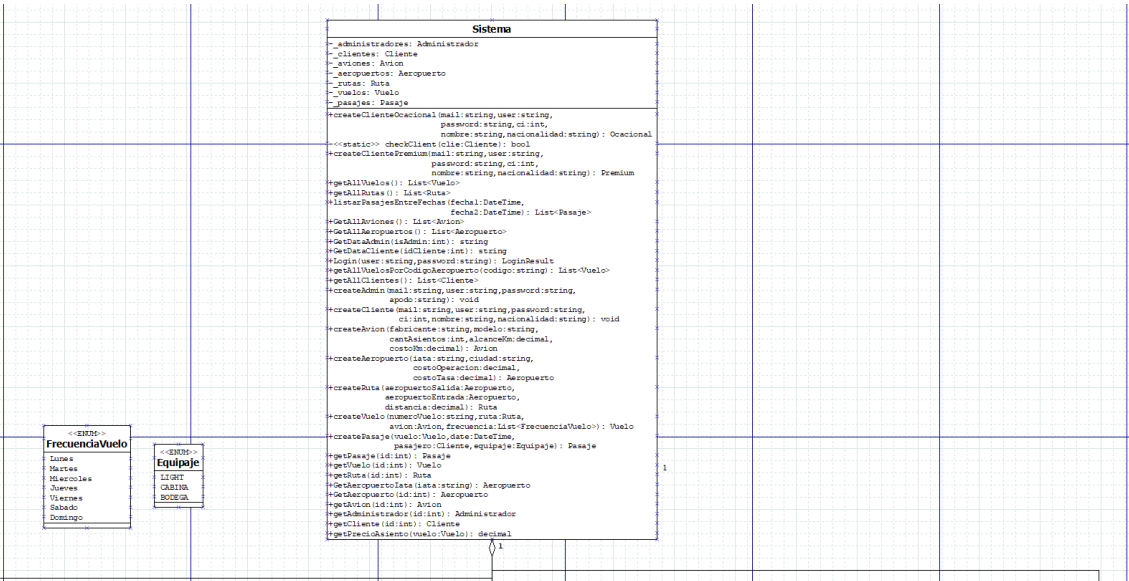
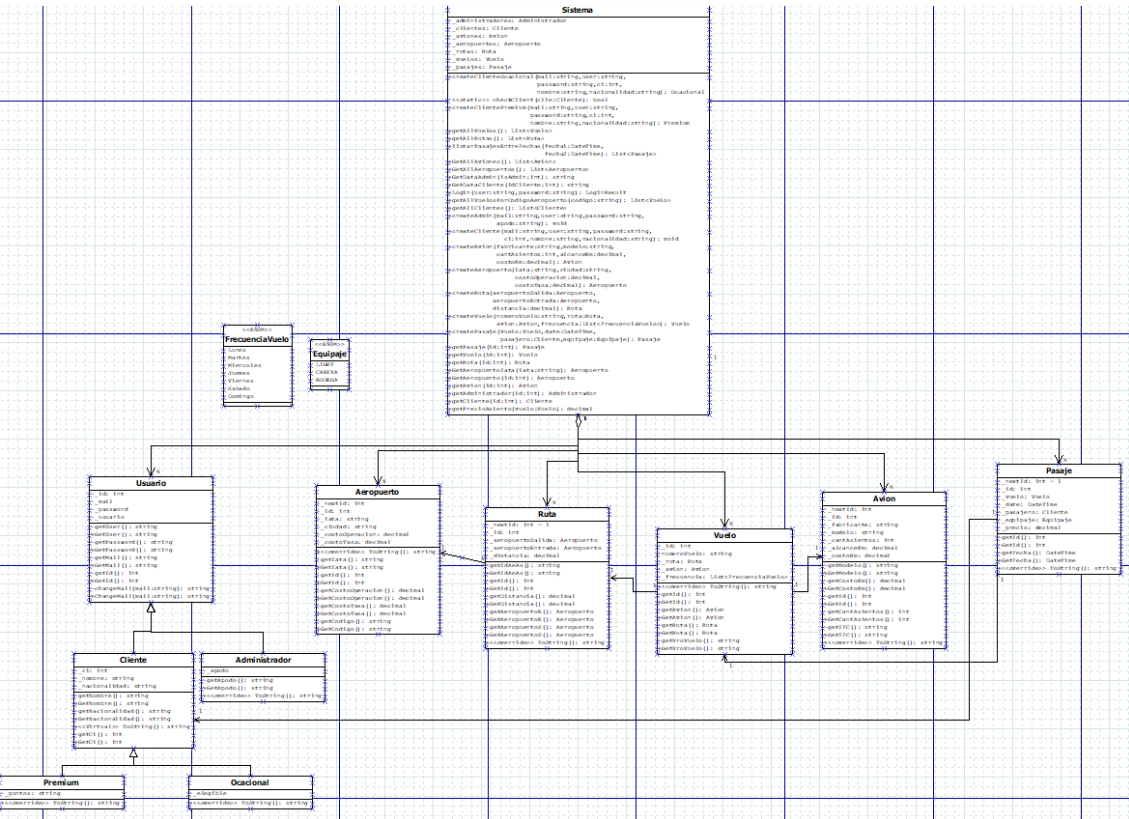
```
system.createPasaje(vuelo1, new DateTime(2004, 09, 10),
pasajero1, Equipaje.BODEGA);
system.createPasaje(vuelo2, new DateTime(2005, 09, 10),
pasajero2, Equipaje.LIGHT);
system.createPasaje(vuelo3, new DateTime(2006, 09, 10),
pasajero3, Equipaje.CABINA);
```

Aquí tienes 25 pasajes creados con los pasajeros proporcionados, usando distintos vuelos, fechas y tipos de equipaje, como en el ejemplo:

```
csharp Copiar Editar

system.createPasaje(vuelo1, new DateTime(1994, 09, 10), pasajero1, Equipaje.BODEGA);
system.createPasaje(vuelo2, new DateTime(1995, 09, 10), pasajero2, Equipaje.LIGHT);
system.createPasaje(vuelo3, new DateTime(1996, 09, 10), pasajero3, Equipaje.CABINA);
system.createPasaje(vuelo4, new DateTime(1997, 01, 15), pasajero4, Equipaje.LIGHT);
system.createPasaje(vuelo5, new DateTime(1998, 02, 20), pasajero5, Equipaje.BODEGA);
system.createPasaje(vuelo6, new DateTime(1999, 04, 25), pasajero6, Equipaje.CABINA);
system.createPasaje(vuelo7, new DateTime(2010, 05, 30), pasajero7, Equipaje.BODEGA);
system.createPasaje(vuelo8, new DateTime(2011, 06, 05), pasajero8, Equipaje.LIGHT);
system.createPasaje(vuelo9, new DateTime(2012, 07, 10), pasajero9, Equipaje.CABINA);
system.createPasaje(vuelo10, new DateTime(2013, 08, 15), pasajero10, Equipaje.BODEGA);
system.createPasaje(vuelo11, new DateTime(2014, 09, 20), pasajero1, Equipaje.LIGHT);
system.createPasaje(vuelo12, new DateTime(2015, 10, 25), pasajero2, Equipaje.CABINA);
system.createPasaje(vuelo13, new DateTime(2016, 11, 30), pasajero3, Equipaje.BODEGA);
```

Diagrama UML





Archivo .DIA en el .ZIP con el diagrama UML

CODIGO

```
//PROGRAM
```

```
using Dominio;
```

```
namespace Obligatorio1
```

```
{
```

```
    internal class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Sistema system = Sistema.Instancia;
```

```
            CrearEjemplos ejemplos = new CrearEjemplos();
```

```
            ejemplos.crearEjemplos(system);
```

```
            int opt = -1;
```

```
            while (opt != 0)
```

```
            {
```

```
                try
```

```
                {
```

```
                    Console.WriteLine("----Menu----");
```

```
                    Console.WriteLine("1-Clientes");
```

```
                    Console.WriteLine("2-Aeropuertos");
```

```
                    Console.WriteLine("3-Pasajes");
```

```
                    Console.WriteLine("0-Salir");
```

```
                    Console.WriteLine("-----");
```

```
    opt = Int32.Parse(Console.ReadLine());  
    Console.Clear();  
    switch (opt)  
    {  
        case 1:  
            Menus.menuCliente(system);  
            break;  
        case 2:  
            Menus.menuAeropuerto(system);  
            break;  
        case 3:  
            Menus.menuPasajes(system);  
  
            break;  
        case 0:  
            Environment.Exit(0);  
            break;  
    }  
  
    }  
    catch (Exception ex)  
    {  
        Console.WriteLine(ex.Message);  
    }  
}  
  
}
```

```

    }
}

//METODOS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

using Dominio;

namespace Obligatorio1
{
    public class Metodos
    {
        public static void isAdmin(bool isAdmin,Sistema system,int user_logged)
        {
            if (isAdmin)
            {
                Console.WriteLine(system.GetDataAdmin(user_logged));
            }
            else
            {
                Console.WriteLine(system.GetDataCliente(user_logged));
            }
            Console.ReadKey();
            Console.Clear();
        }
    }
}

```



```
public static void listarPasajesEntreFechas(Sistema system)
{
    DateTime fecha1;
    DateTime fecha2;
    Console.WriteLine("Ingrese la primera fecha");
    try
    {
        fecha1 = DateTime.Parse(Console.ReadLine());
    }
    catch
    {
        throw new Exception("La fecha ingresada es invalida");
    }
    if (fecha1 == null) throw new Exception("La fecha no es valida");
    Console.WriteLine("Ingrese la segunda fecha");
    try
    {
        fecha2 = DateTime.Parse(Console.ReadLine());
    }
    catch
    {
        throw new Exception("La fecha infresada es invalida");
    }
    if (fecha2 == null) throw new Exception("La fecha no es valida");
    List<Pasaje> newList = system.listarPasajesEntreFechas(fecha1, fecha2);
    Console.Clear();
}
```

```
Console.WriteLine("Pasajes entre fechas\n");  
foreach (Pasaje pasaje in newList1)  
{  
    Console.WriteLine(pasaje.ToString());  
}
```

```
Console.ReadKey();  
Console.Clear();  
}
```

```
public static void getAllClientes(Sistema system)  
{  
    foreach (var item in system.getAllClientes())  
    {  
        Console.WriteLine($"{item.ToString()}\n");  
    }  
    Console.ReadKey();  
    Console.Clear();  
}
```

```
public static void altaCliente(Sistema system)  
{  
    try  
    {  
        Console.WriteLine("Ingrese el mail del cliente");  
        string mail = Console.ReadLine();  
        if (String.IsNullOrEmpty(mail)) throw new Exception("No se ingreso un mail");  
        Console.WriteLine("Ingrese el usuario del cliente");
```

```
string user = Console.ReadLine();

if (String.IsNullOrEmpty(user)) throw new Exception("No se ingreso un
usuario");

Console.WriteLine("Ingrese la password del cliente");

string password = Console.ReadLine();

if (String.IsNullOrEmpty(password)) throw new Exception("No se ingreso una
contraseña");

Console.WriteLine("Ingrese el documento del cliente");

int ci = 0;

try
{
    ci = Int32.Parse(Console.ReadLine());
}
catch
{
    new Exception("Ingrese datos validos");
}

Console.WriteLine("Ingrese el nombre del cliente");

string nombre = Console.ReadLine();

if (String.IsNullOrEmpty(nombre)) throw new Exception("No se ingreso una
conrtaseña");

Console.WriteLine("Ingrese la nacionalidad del cliente");

string nacionalidad = Console.ReadLine();

if (String.IsNullOrEmpty(nacionalidad)) throw new Exception("No se ingreso
una conrtaseña");

Cliente createdCli = system.createClienteOcasional(mail, user, password, ci,
nombre, nacionalidad);

Console.Clear();

Console.WriteLine("Se ha creado satisfactoriamente el cliente!");

Console.ReadKey();
```

```
}  
catch(Exception ex)  
{  
    Console.WriteLine(ex.Message);  
}  
}
```

```
public static void getAllVuelosPorCodigoAeropuerto(Sistema system)  
{  
    Console.WriteLine("Ingrese el codigo IATA");  
    string IATA = Console.ReadLine();  
    if (String.IsNullOrEmpty(IATA)) throw new Exception("No ingreso el codigo  
IATA");  
    foreach (var item in system.getAllVuelosPorCodigoAeropuerto(IATA))  
    {  
        Console.WriteLine(item.ToString());  
    }  
    Console.ReadKey();  
}
```

```
public static void altaAeropuerto(Sistema system)  
{  
    Console.WriteLine("Ingrese un codigo IATA para el aeropuerto.");  
    string clata = Console.ReadLine();  
    Console.WriteLine("Ingrese en que ciudad esta el aeropuerto");  
    string ciudad = Console.ReadLine();
```

```

Console.WriteLine("Ingrese el costo de opracion del aeropuerto");
decimal costoOp = Decimal.Parse(Console.ReadLine());
Console.WriteLine("Ingrese el costo de tasa del aeropuerto");
decimal costoTasa = Decimal.Parse(Console.ReadLine());
system.createAeropuerto(clata, ciudad, costoOp, costoTasa);

Console.Clear();

Console.WriteLine("Se ha creado satisfactoriamente el aeropuerto!");
Console.ReadKey();
}

```

```

public static void getAllAeropuertos(Sistema system)
{
    List<Aeropuerto> aeropuertos = system.GetAllAeropuertos();
    Console.WriteLine("Los aeropuertos actuales son:");
    foreach (var aeropuerto in aeropuertos)
    {
        Console.Write(aeropuerto.ToString());
    }
    Console.Clear();
    Console.ReadKey();
    Console.Clear();
}

```

```

public static void getAllAeropuertosIATA(Sistema system)
{
    List<Aeropuerto> aeropuertos = system.GetAllAeropuertos();
    foreach (var aeropuerto in aeropuertos)

```

```

    {
        Console.WriteLine($"{aeropuerto.GetIata()}\n");
    }

    Console.Clear();
    Console.ReadKey();
    Console.Clear();
}

public static void getAllAviones(Sistema system)
{
    List<Avion> aviones = system.GetAllAviones();
    foreach (var avion in aviones)
    {
        Console.WriteLine(avion.ToString());
    }

    Console.Clear();
    Console.ReadKey();
    Console.Clear();
}

public static void getAllAvionesMinimizado(Sistema system)
{
    List<Avion> aviones = system.GetAllAviones();
    foreach (var avion in aviones)
    {
        Console.WriteLine(avion.GetIFC());
    }
}

```

```
}
```

```
public static void altaAvion(Sistema system)
{
    Console.WriteLine("Fabricante de avion:");
    string fabricante = Console.ReadLine();
    Console.WriteLine("Modelo de avion:");
    string modelo = Console.ReadLine();
    Console.WriteLine("Cantidad de asientos:");
    int cantAsientos = Int32.Parse(Console.ReadLine());
    Console.WriteLine("Alcance de Kilometros:");
    decimal alcanceKm = Decimal.Parse(Console.ReadLine());
    Console.WriteLine("Costo por Kilometro:");
    decimal costoKm = Decimal.Parse(Console.ReadLine());
    system.createAvion(fabricante, modelo, cantAsientos, alcanceKm, costoKm);

    Console.Clear();
    Console.WriteLine("Se ha creado satisfactoriamente el avion!");
    Console.ReadKey();
}
```

```
public static void listarRutas(Sistema system)
{
    Console.Clear();
    List<Ruta> rutas= system.getAllRutas();
    foreach (Ruta ruta in rutas)
```

```

{
    Console.WriteLine(ruta.ToString());
}

Console.Clear();
Console.ReadKey();
Console.Clear();
}

public static void altaRuta(Sistema system)
{
    Console.Clear();
    getAllAeropuertosIATA(system);
    Console.WriteLine("Ingrese el codigo de aeropuerto de salida:\n");
    string iataSalida = Console.ReadLine();
    Console.WriteLine("Ingrese el aeropuerto de salida:\n");
    string iataEntrada = Console.ReadLine();
    Console.WriteLine("Ingrese la distancia entre aeropuertos");
    decimal distancia = Decimal.Parse(Console.ReadLine());

    Aeropuerto aSalida = system.GetAeropuertolata(iataSalida);
    Aeropuerto aEntrada = system.GetAeropuertolata(iataEntrada);

    system.createRuta(aSalida, aEntrada, distancia);

    Console.Clear();
    Console.WriteLine("Se ha creado satisfactoriamente la ruta!");
    Console.ReadKey();
}

```



```
}
```

```
public static void GetAllVuelos(Sistema system)
```

```
{
```

```
    Console.Clear();
```

```
    List<Vuelo> vuelos = system.getAllVuelos();
```

```
    foreach (Vuelo vuelo in vuelos)
```

```
    {
```

```
        Console.WriteLine(vuelo.ToString());
```

```
    }
```

```
    Console.ReadKey();
```

```
    Console.Clear();
```

```
}
```

```
public static void listarRutaMinimizado(Sistema system)
```

```
{
```

```
    Console.Clear();
```

```
    List<Ruta> rutas = system.getAllRutas();
```

```
    foreach (Ruta ruta in rutas)
```

```
    {
```

```
        Console.WriteLine(ruta.GetIdAeAs());
```

```
    }
```

```
}
```

```
public static void altaVuelo(Sistema system)
```

```
{
```

```
    Console.Clear();
```

```
Console.WriteLine("Ingrese el numero de vuelo");
```

```
string numeroVuelo = Console.ReadLine();
```

```
Console.WriteLine("Ingrese el id de la ruta que seguira el vuelo:");
```

```
listarRutaMinimizado(system);
```

```
int idRuta = Int32.Parse(Console.ReadLine());
```

```
Ruta ruta = system.getRuta(idRuta);
```

```
Console.Clear();
```

```
Console.WriteLine("Ingrese id del avion que hara el vuelo");
```

```
getAllAvionesMinimizado(system);
```

```
int idAvion = Int32.Parse(Console.ReadLine());
```

```
Console.Clear();
```

```
Avion avion = system.getAvion(idAvion);
```

```
Console.WriteLine("Ingrese la frecuencia que se hara el vuelo\n");
```

```
Console.WriteLine("1-Lunes");
```

```
Console.WriteLine("2-Martes");
```

```
Console.WriteLine("3-Miercoles");
```

```
Console.WriteLine("4-Jueves");
```

```
Console.WriteLine("5-Viernes");
```

```
Console.WriteLine("6-Sabado");
```

```
Console.WriteLine("7-Domingo");
```

```
Console.WriteLine("0-Terminar");
```

```

List<FrecuenciaVuelo> frecuencias = new List<FrecuenciaVuelo>();

int frecuencia = -1;

while (frecuencia != 0)
{
    frecuencia = Int32.Parse(Console.ReadLine());
    if (frecuencia == 1) frecuencias.Add(FrecuenciaVuelo.Lunes);
    if (frecuencia == 2) frecuencias.Add(FrecuenciaVuelo.Martes);
    if (frecuencia == 3) frecuencias.Add(FrecuenciaVuelo.Miercoles);
    if (frecuencia == 4) frecuencias.Add(FrecuenciaVuelo.Jueves);
    if (frecuencia == 5) frecuencias.Add(FrecuenciaVuelo.Viernes);
    if (frecuencia == 6) frecuencias.Add(FrecuenciaVuelo.Sabado);
    if (frecuencia == 7) frecuencias.Add(FrecuenciaVuelo.Domingo);
}

system.createVuelo(numeroVuelo,ruta,avion,frecuencias);

Console.Clear();

Console.WriteLine("Se ha creado satisfactoriamente el vuelo!");

Console.ReadKey();

}

}

}

```



```
}  
}
```

```
public static void menuAeropuerto(Sistema system)  
{  
    Console.Clear();  
    Console.WriteLine("1-Conseguir todos los vuelos de un aeropuerto");  
    int opt = Int32.Parse(Console.ReadLine());  
  
    switch (opt)  
    {  
        case 1:  
            Metodos.getAllVuelosPorCodigoAeropuerto(system);  
            break;  
    }  
}
```

```
public static void menuAviones(Sistema system)  
{  
    Console.Clear();  
    Console.WriteLine("1-Listar todos los aviones");  
    Console.WriteLine("2-Crear un avion");  
    int opt = Int32.Parse(Console.ReadLine());  
  
    switch (opt)  
    {  
        case 1:  
            Metodos.getAllAviones(system);
```

```
break;
```

```
case 2:
```

```
Metodos.altaAvion(system);
```

```
break;
```

```
}
```

```
}
```

```
public static void menuPasajes(Sistema system)
```

```
{
```

```
Console.Clear();
```

```
Console.WriteLine("1-Listar pasajes entre dos fechas");
```

```
int opt = Int32.Parse(Console.ReadLine());
```

```
switch (opt)
```

```
{
```

```
case 1:
```

```
Metodos.listarPasajesEntreFechas(system);
```

```
break;
```

```
}
```

```
}
```

```
public static void menuRutas(Sistema system)
```

```
{
```

```
Console.Clear();
```

```
Console.WriteLine("1-Listar todas las rutas");
```

```
Console.WriteLine("2-Crear una ruta");
```

```

int opt = Int32.Parse(Console.ReadLine());

switch (opt)
{
    case 1:
        Metodos.listarRutas(system);
        break;
    case 2:
        Metodos.altaRuta(system);
        break;
}
}

public static void menuVuelos(Sistema system)
{
    Console.Clear();
    Console.WriteLine("1-Listar todos los vuelos");
    Console.WriteLine("2-Crear un vuelo");
    int opt = Int32.Parse(Console.ReadLine());
    switch (opt)
    {
        case 1:
            Metodos.GetAllVuelos(system);
            break;
        case 2:
            Metodos.altaVuelo(system);
            break;
    }
}
}

```

```
}  
}
```

```
//CREAR EJEMPLOS
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
using Dominio;
```

```
namespace Obligatorio1
```

```
{
```

```
    public class CrearEjemplos
```

```
    {
```

```
        public void crearEjemplos(Sistema system)
```

```
        {
```

```
            //creacion de admin y cliente
```

```
            system.createAdmin("joaco@zionbancorp.com", "joaco", "pass",  
"joacoApodo");
```

```
            system.createAdmin("isma@microsoft.com", "isma", "pass", "ismaApodo");
```

```
            //CREA OCACIONALES
```

```
            Cliente pasajero1 =
```

```
system.createClienteOccasional("ezequieltitan@hotmail.com", "eze", "pass", 46689505,  
"Ezequiel", "Uruguayo");
```

```
            Cliente pasajero2 = system.createClienteOccasional("sofia@mail.com", "sofi",  
"1234", 45321231, "Sofia", "Uruguay");
```

```
            Cliente pasajero3 = system.createClienteOccasional("marco@mail.com",  
"marco123", "abcd", 48976543, "Marco", "Chileno");
```



```
Cliente pasajero4 = system.createClienteOccasional("lucia@mail.com", "lucy",
"qwerty", 45511222, "Lucia", "Paraguaya");

Cliente pasajero5 = system.createClienteOccasional("diego@mail.com",
"dieguito", "zxcv", 44112233, "Diego", "Boliviano");

//CREA PREMIUM

Cliente pasajero6 = system.createClientePremium("estimail@premium.com",
"esti", "pass", 12332126, "Esteban", "Argentino");

Cliente pasajero7 = system.createClientePremium("ana@mail.com", "anita",
"anapass", 99887766, "Ana", "Uruguaya");

Cliente pasajero8 = system.createClientePremium("jose@mail.com", "jose",
"josito", 11223344, "Jose", "Peruano");

Cliente pasajero9 = system.createClientePremium("carla@mail.com", "carli",
"car123", 55667788, "Carla", "Venezolana");

Cliente pasajero10 = system.createClientePremium("nicolas@mail.com", "nico",
"nicopass", 77889900, "Nicolas", "Colombiano");

//creacion de aeropuerto

Aeropuerto aeropuertoUYU = system.createAeropuerto("UYU", "Montevideo",
80, 17);

Aeropuerto aeropuertoARG = system.createAeropuerto("ARG", "Buenos Aires",
120, 12);

Aeropuerto aeropuertoUSA = system.createAeropuerto("USA", "New York", 77,
24);

Aeropuerto aeropuertoNIG = system.createAeropuerto("NIG", "Lagos", 33, 6);

Aeropuerto aeropuertoBRA = system.createAeropuerto("BRA", "Rio de Janeiro",
95, 15);

Aeropuerto aeropuertoMEX = system.createAeropuerto("MEX", "Ciudad de
México", 85, 13);

Aeropuerto aeropuertoCAN = system.createAeropuerto("CAN", "Toronto", 105,
18);

Aeropuerto aeropuertoFRA = system.createAeropuerto("FRA", "Frankfurt", 150,
22);

Aeropuerto aeropuertoESP = system.createAeropuerto("ESP", "Madrid", 110,
14);
```

```

Aeropuerto aeropuertoITA = system.createAeropuerto("ITA", "Roma", 100, 16);
Aeropuerto aeropuertoCHL = system.createAeropuerto("CHL", "Santiago", 90,
11);

Aeropuerto aeropuertoPER = system.createAeropuerto("PER", "Lima", 88, 10);
Aeropuerto aeropuertoCOL = system.createAeropuerto("COL", "Bogotá", 87, 9);
Aeropuerto aeropuertoVEN = system.createAeropuerto("VEN", "Caracas", 70, 8);
Aeropuerto aeropuertoURU = system.createAeropuerto("URU", "Punta del
Este", 60, 7);

Aeropuerto aeropuertoUK = system.createAeropuerto("UK", "Londres", 140,
25);

Aeropuerto aeropuertoJPN = system.createAeropuerto("JPN", "Tokio", 160, 30);
Aeropuerto aeropuertoCHN = system.createAeropuerto("CHN", "Pekín", 155,
28);

Aeropuerto aeropuertoAUS = system.createAeropuerto("AUS", "Sídney", 145,
20);

Aeropuerto aeropuertoIND = system.createAeropuerto("IND", "Nueva Delhi",
130, 19);

//creacion de ruta
Ruta rutaUyuArg = system.createRuta(aeropuertoUYU, aeropuertoARG, 1000);
Ruta rutaUyuBra = system.createRuta(aeropuertoUYU, aeropuertoBRA, 1800);
Ruta rutaUyuChl = system.createRuta(aeropuertoUYU, aeropuertoCHL, 1600);
Ruta rutaArgUsa = system.createRuta(aeropuertoARG, aeropuertoUSA, 8500);
Ruta rutaArgFra = system.createRuta(aeropuertoARG, aeropuertoFRA, 11000);
Ruta rutaArgEsp = system.createRuta(aeropuertoARG, aeropuertoESP, 10200);
Ruta rutaBraMex = system.createRuta(aeropuertoBRA, aeropuertoMEX, 7400);
Ruta rutaBraNig = system.createRuta(aeropuertoBRA, aeropuertoNIG, 6800);
Ruta rutaChlPer = system.createRuta(aeropuertoCHL, aeropuertoPER, 2500);
Ruta rutaChlCol = system.createRuta(aeropuertoCHL, aeropuertoCOL, 2700);
Ruta rutaMexCan = system.createRuta(aeropuertoMEX, aeropuertoCAN, 3600);
Ruta rutaCanUsa = system.createRuta(aeropuertoCAN, aeropuertoUSA, 800);

```

```
Ruta rutaUsaFra = system.createRuta(aeropuertoUSA, aeropuertoFRA, 5900);
Ruta rutaUsaEsp = system.createRuta(aeropuertoUSA, aeropuertoESP, 6100);
Ruta rutaUsalta = system.createRuta(aeropuertoUSA, aeropuertoITA, 7000);
Ruta rutaFraUk = system.createRuta(aeropuertoFRA, aeropuertoUK, 900);
Ruta rutaEspUk = system.createRuta(aeropuertoESP, aeropuertoUK, 1300);
Ruta rutaUkJpn = system.createRuta(aeropuertoUK, aeropuertoJPN, 9700);
Ruta rutaUkChn = system.createRuta(aeropuertoUK, aeropuertoCHN, 8900);
Ruta rutaJpnAus = system.createRuta(aeropuertoJPN, aeropuertoAUS, 7800);
Ruta rutaChnInd = system.createRuta(aeropuertoCHN, aeropuertoIND, 4800);
Ruta rutaIndAus = system.createRuta(aeropuertoIND, aeropuertoAUS, 10100);
Ruta rutaIndNig = system.createRuta(aeropuertoIND, aeropuertoNIG, 7500);
Ruta rutaNigVen = system.createRuta(aeropuertoNIG, aeropuertoVEN, 9200);
Ruta rutaVenCol = system.createRuta(aeropuertoVEN, aeropuertoCOL, 1600);
Ruta rutaColPer = system.createRuta(aeropuertoCOL, aeropuertoPER, 1800);
Ruta rutaPerEsp = system.createRuta(aeropuertoPER, aeropuertoESP, 9600);
Ruta rutaCanFra = system.createRuta(aeropuertoCAN, aeropuertoFRA, 6000);
Ruta rutaUyuUk = system.createRuta(aeropuertoUYU, aeropuertoUK, 11400);
Ruta rutaEspJpn = system.createRuta(aeropuertoESP, aeropuertoJPN, 10800);
```

```
//creacion de avion
```

```
Avion primerAvion = system.createAvion("Nissan", "modeloJapones", 200, 50,
5);
```

```
Avion segundoAvion = system.createAvion("Boeing", "737 Max", 180, 6500,
8.5m);
```

```
Avion tercerAvion = system.createAvion("Airbus", "A320neo", 190, 6100, 8.0m);
```

```
Avion cuartoAvion = system.createAvion("Embraer", "E195-E2", 132, 4800,
6.5m);
```

//creacion de vuelo

Vuelo vuelo1 = system.createVuelo("AA334", rutaUyuArg, primerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes, FrecuenciaVuelo.Martes });

Vuelo vuelo2 = system.createVuelo("BB101", rutaUyuBra, segundoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles, FrecuenciaVuelo.Viernes });

Vuelo vuelo3 = system.createVuelo("CC202", rutaUyuChl, tercerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Sabado });

Vuelo vuelo4 = system.createVuelo("DD303", rutaArgUsa, cuartoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });

Vuelo vuelo5 = system.createVuelo("EE404", rutaArgFra, segundoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Jueves });

Vuelo vuelo6 = system.createVuelo("FF505", rutaArgEsp, tercerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Viernes });

Vuelo vuelo7 = system.createVuelo("GG606", rutaBraMex, cuartoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });

Vuelo vuelo8 = system.createVuelo("HH707", rutaBraNig, primerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles, FrecuenciaVuelo.Sabado });

Vuelo vuelo9 = system.createVuelo("II808", rutaChlPer, segundoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });

Vuelo vuelo10 = system.createVuelo("JJ909", rutaChlCol, tercerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });

Vuelo vuelo11 = system.createVuelo("KK111", rutaMexCan, cuartoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles, FrecuenciaVuelo.Jueves });

Vuelo vuelo12 = system.createVuelo("LL222", rutaCanUsa, primerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Viernes });

Vuelo vuelo13 = system.createVuelo("MM333", rutaUsaFra, segundoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Jueves });

Vuelo vuelo14 = system.createVuelo("NN444", rutaUsaEsp, tercerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Sabado });

Vuelo vuelo15 = system.createVuelo("OO555", rutaUsalta, cuartoAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });

Vuelo vuelo16 = system.createVuelo("PP666", rutaFraUk, primerAvion, new List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });

```
Vuelo vuelo17 = system.createVuelo("QQ777", rutaEspUk, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo18 = system.createVuelo("RR888", rutaUkJpn, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Viernes, FrecuenciaVuelo.Sabado });
```

```
Vuelo vuelo19 = system.createVuelo("SS999", rutaUkChn, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles });
```

```
Vuelo vuelo20 = system.createVuelo("TT000", rutaJpnAus, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes });
```

```
Vuelo vuelo21 = system.createVuelo("UU121", rutaChnInd, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes });
```

```
Vuelo vuelo22 = system.createVuelo("VV232", rutaIndAus, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });
```

```
Vuelo vuelo23 = system.createVuelo("WW343", rutaIndNig, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Miercoles });
```

```
Vuelo vuelo24 = system.createVuelo("XX454", rutaNigVen, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo25 = system.createVuelo("YY565", rutaVenCol, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Sabado });
```

```
Vuelo vuelo26 = system.createVuelo("ZZ676", rutaColPer, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Lunes, FrecuenciaVuelo.Miercoles });
```

```
Vuelo vuelo27 = system.createVuelo("AB787", rutaPerEsp, cuartoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Viernes });
```

```
Vuelo vuelo28 = system.createVuelo("BC898", rutaCanFra, primerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Domingo });
```

```
Vuelo vuelo29 = system.createVuelo("CD909", rutaUyuUk, segundoAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Martes, FrecuenciaVuelo.Jueves });
```

```
Vuelo vuelo30 = system.createVuelo("DE010", rutaEspJpn, tercerAvion, new  
List<FrecuenciaVuelo> { FrecuenciaVuelo.Sabado });
```

```
//conseguir precio por asiento
```

```
//system.getPrecioAsiento(vuelo1);
```

//creacion de pasaje

Pasaje pasaje1 = system.createPasaje(vuelo1, new DateTime(2003, 09, 10),
pasajero1, Equipaje.BODEGA);

Pasaje pasaje2 = system.createPasaje(vuelo2, new DateTime(2004, 10, 12),
pasajero2, Equipaje.LIGHT);

Pasaje pasaje3 = system.createPasaje(vuelo3, new DateTime(2005, 11, 14),
pasajero3, Equipaje.CABINA);

Pasaje pasaje4 = system.createPasaje(vuelo4, new DateTime(2006, 12, 16),
pasajero4, Equipaje.LIGHT);

Pasaje pasaje5 = system.createPasaje(vuelo5, new DateTime(2007, 01, 18),
pasajero5, Equipaje.BODEGA);

Pasaje pasaje6 = system.createPasaje(vuelo6, new DateTime(2008, 02, 20),
pasajero6, Equipaje.CABINA);

Pasaje pasaje7 = system.createPasaje(vuelo7, new DateTime(2009, 03, 22),
pasajero7, Equipaje.BODEGA);

Pasaje pasaje8 = system.createPasaje(vuelo8, new DateTime(2010, 04, 24),
pasajero8, Equipaje.LIGHT);

Pasaje pasaje9 = system.createPasaje(vuelo9, new DateTime(2011, 05, 26),
pasajero9, Equipaje.CABINA);

Pasaje pasaje10 = system.createPasaje(vuelo10, new DateTime(2012, 06, 28),
pasajero10, Equipaje.BODEGA);

Pasaje pasaje11 = system.createPasaje(vuelo11, new DateTime(2013, 07, 30),
pasajero1, Equipaje.LIGHT);

Pasaje pasaje12 = system.createPasaje(vuelo12, new DateTime(2014, 08, 02),
pasajero2, Equipaje.CABINA);

Pasaje pasaje13 = system.createPasaje(vuelo13, new DateTime(2015, 09, 04),
pasajero3, Equipaje.BODEGA);

Pasaje pasaje14 = system.createPasaje(vuelo14, new DateTime(2016, 10, 06),
pasajero4, Equipaje.LIGHT);

Pasaje pasaje15 = system.createPasaje(vuelo15, new DateTime(2017, 11, 08),
pasajero5, Equipaje.CABINA);

Pasaje pasaje16 = system.createPasaje(vuelo16, new DateTime(2018, 12, 10),
pasajero6, Equipaje.BODEGA);

```
Pasaje pasaje17 = system.createPasaje(vuelo17, new DateTime(2019, 01, 12),  
pasajero7, Equipaje.LIGHT);
```

```
Pasaje pasaje18 = system.createPasaje(vuelo18, new DateTime(2020, 02, 14),  
pasajero8, Equipaje.CABINA);
```

```
Pasaje pasaje19 = system.createPasaje(vuelo19, new DateTime(2021, 03, 16),  
pasajero9, Equipaje.BODEGA);
```

```
Pasaje pasaje20 = system.createPasaje(vuelo20, new DateTime(2022, 04, 18),  
pasajero10, Equipaje.LIGHT);
```

```
Pasaje pasaje21 = system.createPasaje(vuelo21, new DateTime(2023, 05, 20),  
pasajero1, Equipaje.CABINA);
```

```
Pasaje pasaje22 = system.createPasaje(vuelo22, new DateTime(2024, 06, 22),  
pasajero2, Equipaje.BODEGA);
```

```
Pasaje pasaje23 = system.createPasaje(vuelo23, new DateTime(2025, 07, 24),  
pasajero3, Equipaje.LIGHT);
```

```
Pasaje pasaje24 = system.createPasaje(vuelo24, new DateTime(2026, 08, 26),  
pasajero4, Equipaje.CABINA);
```

```
Pasaje pasaje25 = system.createPasaje(vuelo25, new DateTime(2027, 09, 28),  
pasajero5, Equipaje.BODEGA);
```

```
}
```

```
}
```

```
}
```

```
//SISTEMA
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Text;
```

```
namespace Dominio
```

```
{
```

```
    public class Sistema
```

```
{
```

//UML HECHO

private static Sistema _instancia;

public static Sistema Instancia

{

get {

if(_instancia == null) _instancia = new Sistema();

return _instancia;

}

}

private Sistema() { }

private List<Administrador> _administradores = new List<Administrador>();

private List<Cliente> _clientes = new List<Cliente>();

private List<Avion> _aviones = new List<Avion>();

private List<Aeropuerto> _aeropuertos = new List<Aeropuerto>();

private List<Ruta> _rutas = new List<Ruta>();

private List<Vuelo> _vuelos = new List<Vuelo>();

private List<Pasaje> _pasajes = new List<Pasaje>();

public Ocasional createClienteOcasional(string mail, string user, string password,
int ci, string nombre, string nacionalidad)

{

Ocasional cli = new Ocasional(mail, user, password, ci, nombre, nacionalidad);

if (checkClient(cli,_clientes))

{

_clientes.Add(cli);


```
}  
    return cli;  
}
```

```
private static bool checkClient(Cliente clie, List<Cliente> clientes)  
{  
    foreach (Cliente cl in clientes)  
    {  
        if (cl.GetCi() == clie.GetCi()) throw new Exception("La cedula ya esta  
registrada.");  
        if (cl.GetMail() == clie.GetMail()) throw new Exception("El mail ya esta  
registrado");  
    }  
    if (clie.GetCi().ToString().Length != 8) throw new Exception("La cantidad de  
caracteres en la cedula debe de ser 8.");  
    if (String.IsNullOrEmpty(clie.GetMail())) throw new Exception("Ningun mail fue  
introducido");  
    if (!clie.GetMail().Contains("@")) throw new Exception("El mail debe contener  
un arroba");  
    if (String.IsNullOrEmpty(clie.GetNombre())) throw new Exception("Ningun  
nombre fue introducido");  
    if (String.IsNullOrEmpty(clie.GetNacionalidad())) throw new Exception("Ninguna  
nacionalidad fue introducida");  
    if (String.IsNullOrEmpty(clie.GetUser())) throw new Exception("Ningun User fue  
introducido");  
    if (String.IsNullOrEmpty(clie.GetPassword())) throw new Exception("Ninguna  
contraseña fue introducida");  
    return true;  
}
```

```
public Premium createClientePremium(string mail, string user, string password, int
ci, string nombre, string nacionalidad)
```

```
{
    Premium cli = new Premium(mail, user, password, ci, nombre, nacionalidad);
    if (checkClient(cli, _clientes))
    {
        _clientes.Add(cli);
    }
    return cli;
}
```

```
public List<Vuelo> getAllVuelos()
```

```
{
    return _vuelos;
}
```

```
public List<Ruta> getAllRutas()
```

```
{
    return _rutas;
}
```

```
public List<Pasaje> listarPasajesEntreFechas(DateTime fecha1, DateTime fecha2)
```

```
{
    List<Pasaje> listadoNew = new List<Pasaje>();
    foreach (Pasaje pasaje in _pasajes)
    {
        if (fecha1.CompareTo(pasaje.GetFecha()) <= 0 &&
fecha2.CompareTo(pasaje.GetFecha()) >=0)
        {
            listadoNew.Add(pasaje);
        }
    }
}
```

```
    }  
}  
return listadoNew;  
}
```

```
public List<Avion> GetAllAviones()  
{  
    return _aviones;  
}
```

```
public List<Aeropuerto> GetAllAeropuertos()  
{  
    return _aeropuertos;  
}
```

```
public string GetDataAdmin(int idAdmin)  
{  
    foreach (var admin in _administradores)  
    {  
        if(admin.Id == idAdmin)  
        {  
            return admin.ToString();  
        }  
    }  
    return "ERROR al encontrar administrador";  
}
```

```
public string GetDataCliente(int idCliente)
```

```

{
    foreach (var cliente in _clientes)
    {
        if (cliente.Id == idCliente)
        {
            return cliente.ToString();
        }
    }
    return "ERROR al encontrar el cliente";
}

public List<Vuelo> getAllVuelosPorCodigoAeropuerto(string codigo)
{
    List<Vuelo> vuelosNew = new List<Vuelo>();
    foreach (Vuelo vuelo in _vuelos)
    {
        if(vuelo.GetRuta().GetAeropuertoE().GetCodigo().ToLower() ==
codigo.ToLower())
        {
            vuelosNew.Add(vuelo);
        }
        if(vuelo.GetRuta().GetAeropuertoS().GetCodigo().ToLower() ==
codigo.ToLower())
        {
            vuelosNew.Add(vuelo);
        }
    }
    return vuelosNew;
}

```

```
public List<Cliente> getAllClientes()
{
    return _clientes;
}
```

```
public void createAdmin(string mail, string user,string password,string apodo)
{
    Administrador adm = new Administrador(mail,user, password, apodo);
    this._administradores.Add(adm);
}
```

```
public void createCliente(string mail,string user,string password, int ci, string
nombre, string nacionalidad)
{
    Cliente cli = new Cliente(mail,user,password,ci,nombre,nacionalidad);
    _clientes.Add(cli);
}
```

```
public Avion createAvion(string fabricante, string modelo, int cantAsientos,
decimal alcanceKm, decimal costoKm)
{
    Avion av = new Avion(fabricante,modelo,cantAsientos,alcanceKm, costoKm);
    _aviones.Add(av);
    return av;
}
```

```
public Aeropuerto createAeropuerto(string iata, string ciudad, decimal
costoOperacion, decimal costoTasa)
```

```

{
    Aeropuerto aeropuerto = new
Aeropuerto(iata,ciudad, costoOperacion, costoTasa);
    _aeropuertos.Add(aeropuerto);
    return aeropuerto;
}

```

```

public Ruta createRuta(Aeropuerto aeropuertoSalida, Aeropuerto
aeropuertoEntrada, decimal distancia)

```

```

{
    Ruta ruta = new Ruta(aeropuertoSalida,aeropuertoEntrada,distancia);
    _rutas.Add(ruta);
    return ruta;
}

```

```

public Vuelo createVuelo(string numeroVuelo, Ruta ruta, Avion avion,
List<FrecuenciaVuelo> frecuencia)

```

```

{
    Vuelo vuelo = new Vuelo(numeroVuelo, ruta, avion, frecuencia);
    _vuelos.Add(vuelo);
    return vuelo;

}

```

```

public Pasaje createPasaje(Vuelo vuelo, DateTime date, Cliente pasajero, Equipaje
equipaje)

```

```

{
    decimal precioAsiento = getPrecioAsiento(vuelo);
    decimal costo = precioAsiento + precioAsiento * 0.25m;

```

```

if(equipaje == Equipaje.CABINA)
{
    costo = costo + costo * 0.10m;
}
else if(equipaje == Equipaje.BODEGA)
{
    if (pasajero is Premium)
    {
        costo = costo + costo * 0.05m;
    }
    else if (pasajero is Ocacional)
    {
        costo = costo + costo * 0.20m;
    }
}

```

```

decimal costoTasaE = vuelo.GetRuta().GetAeropuertoE().GetCostoTasa();

```

```

decimal costoTasaS = vuelo.GetRuta().GetAeropuertoS().GetCostoTasa();

```

```

costo += costoTasaE + costoTasaS;

```

```

Pasaje pasaje = new Pasaje(vuelo, date, pasajero, equipaje, costo);

```

```

_pasajes.Add(pasaje);

```

```

return pasaje;

```

```

}

```

```

public Pasaje getPasaje(int id)

```

```

{

```

```

    foreach (var pasaje in _pasajes)

```

```
{  
    if (pasaje.GetId() == id)  
    {  
        return pasaje;  
    }  
}  
return null;  
}
```

```
public Vuelo getVuelo(int id)  
{  
    foreach (var vuelo in _vuelos)  
    {  
        if (vuelo.GetId() == id)  
        {  
            return vuelo;  
        }  
    }  
    return null;  
}
```

```
public Ruta getRuta(int id)  
{  
    foreach (var ruta in _rutas)  
    {  
        if (ruta.GetId() == id)  
        {  
            return ruta;  
        }  
    }  
}
```



```
    }  
}  
return null;  
}
```

```
public Aeropuerto GetAeropuertolata(string iata)  
{  
    foreach (Aeropuerto aeropuerto in _aeropuertos)  
    {  
        if(aeropuerto.GetIata() == iata)  
        {  
            return aeropuerto;  
        }  
    }  
    return null;  
}
```

```
public Aeropuerto GetAeropuerto(int id)  
{  
    foreach (var aeropuerto in _aeropuertos)  
    {  
        if (aeropuerto.GetId() == id)  
        {  
            return aeropuerto;  
        }  
    }  
    return null;  
}
```

```
public Avion getAvion(int id)
{
    foreach (var avion in _aviones)
    {
        if(avion.GetId() == id)
        {
            return avion;
        }
    }
    return null;
}
```

```
public Administrador getAdministrador(int id)
{
    foreach (var adm in _administradores)
    {
        if(adm.getId() == id)
        {
            return adm;
        }
    }
    return null;
}
```

```
public Cliente getCliente(int id)
{
    foreach (var cli in _clientes)
```

```
{  
    if (cli.getId() == id)  
    {  
        return cli;  
    }  
}  
return null;  
}
```

```
public decimal getPrecioAsiento(Vuelo vuelo)  
{  
    //(costoKm * distanciaRuta + costoOperacion + costoOperacion )/cantAsientos =  
precio por asiento
```

```
Avion avion = vuelo.GetAvion();  
decimal costoKm = avion.GetCostoKm();
```

```
Ruta rutaVuelo = vuelo.GetRuta();  
decimal distancia = rutaVuelo.GetDistancia();
```

```
Aeropuerto aeropuertoE = rutaVuelo.GetAeropuertoE();  
Aeropuerto aeropuertoS = rutaVuelo.GetAeropuertoS();
```

```
decimal costoAeropuertoE = aeropuertoE.GetCostoOperacion();  
decimal costoAeropuertoS = aeropuertoS.GetCostoOperacion();
```

```
int cantidadAsientos = avion.GetCantAsientos();
```

```
        decimal total = (costoKm * distancia + costoAeropuertoE + costoAeropuertoS) /  
cantidadAsientos;
```

```
        return total;
```

```
    }
```

```
}
```

```
}
```

```
//Vuelo
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Text;
```

```
namespace Dominio
```

```
{
```

```
    public class Vuelo
```

```
    {
```

```
        //UML CREADO
```

```
        private static int _nextId=1;
```

```
        private int _id;
```

```
        private string _numeroVuelo;
```

```
        private Ruta _ruta;
```

```
        private Avion _avion;
```

```
        private List<FrecuenciaVuelo> _frecuencia;
```

```
        public string NumeroVuelo { get { return _numeroVuelo; } set { this._numeroVuelo  
= value; } }
```

```
        public Ruta Ruta { get { return this._ruta; } set {this._ruta = value; } }
```

```
        public Avion Avion { get { return this._avion; } set { this._avion = value; } }
```

```
    public List<FrecuenciaVuelo> Frecuencia { get { return this._frecuencia; } set {  
this._frecuencia = value; } }
```

```
    public Vuelo(string numeroVuelo, Ruta ruta, Avion avion, List<FrecuenciaVuelo>  
frecuencia)
```

```
    {  
        this._id = _nextId++;  
        this._numeroVuelo=numeroVuelo;  
        this._ruta=ruta;  
        this._avion=avion;  
        this._frecuencia=frecuencia;  
    }
```

```
    public Vuelo() { }
```

```
    public override string ToString()
```

```
    {  
        string frecuencia = "";  
        foreach (FrecuenciaVuelo frecuece in this._frecuencia)  
        {  
            frecuencia += frecuece;  
  
            if (frecuece != this._frecuencia[this._frecuencia.Count-1])  
            {  
                frecuencia += "-";  
            }  
        }  
    }
```

```
        return $"Numero de  
vuelo:{this._numeroVuelo}\nModelo:{this._avion.GetModelo()}\nRuta:{this._ruta.GetA
```

```
eropuertoE().GetCodigo()}-  
{this._ruta.GetAeropuertoS().GetCodigo()}\nFrecuencia:{frecuencia}\n";  
}
```

```
private int getId()
```

```
{
```

```
    return this._id;
```

```
}
```

```
public int GetId()
```

```
{
```

```
    return getId();
```

```
}
```

```
private Avion getAvion()
```

```
{
```

```
    return this._avion;
```

```
}
```

```
public Avion GetAvion()
```

```
{
```

```
    return getAvion();
```

```
}
```

```
private Ruta getRuta()
```

```
{
```

```
    return this._ruta;
```

```
}
```

```
public Ruta GetRuta()
```

```
{
```

```
    return getRuta();
```

```

    }

    private string getNroVuelo()
    {
        return this._numeroVuelo;
    }

    public string GetNroVuelo()
    {
        return getNroVuelo();
    }

```

```

    }

```

```

}

```

```

//Usuario

```

```

using System;

```

```

using System.Collections.Generic;

```

```

using System.Text;

```

```

namespace Dominio

```

```

{

```

```

    public class Usuario

```

```

    {

```

```

        //UML hecho

```

```

        private static int _nextId = 1;

```

```

        private int _id;

```

```

        private string _mail;

```

```

        private string _password;

```

```
private string _usuario;
```

```
public int Id { get { return _id; } }
```

```
public string Mail { get { return _mail; } set { _mail = value; } }
```

```
public string User { get { return _usuario; } set { _usuario = value; } }
```

```
public string Password { get { return _password; } set { _password = value; } }
```

```
public Usuario(string mail,string user,string password)
```

```
{
```

```
    this._id = _nextId++;
```

```
    this._mail = mail;
```

```
    this._usuario = user;
```

```
    this._password = password;
```

```
}
```

```
public Usuario() { }
```

```
private string getUser()
```

```
{
```

```
    return this._usuario;
```

```
}
```

```
public string GetUser()
```

```
{
```

```
    return getUser();
```

```
}
```

```
private string getPassword()
```

```
{
```

```
    return this._password;
```



```
}
```

```
public string GetPassword()
```

```
{
```

```
    return getPassword();
```

```
}
```

```
private string getMail()
```

```
{
```

```
    return this._mail;
```

```
}
```

```
public string GetMail()
```

```
{
```

```
    return getMail();
```

```
}
```

```
public int getId()
```

```
{
```

```
    return this._id;
```

```
}
```

```
public int GetId()
```

```
{
```

```
    return getId();
```

```
}
```

```
private void changeMail(string mail)
```

```
{
```

```
    this._mail = mail;
```

```
}
```

```

        public void ChangeMail(string mail)
        {
            changeMail(mail);
        }
    }
}

//Ruta
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{

    public class Ruta
    {
        //UML CREADO
        private static int _idNext=1;

        private int _id;
        private Aeropuerto _aeropuertoSalida;
        private Aeropuerto _aeropuertoEntrada;
        private decimal _distancia;

        public int Id { get { return _id; } }

        private string getIdAeAs ()

```

```

{
    string iataE = this._aeropuertoEntrada.GetIata();
    string iataS = this._aeropuertoSalida.GetIata();
    return $"ID:{this._id} | {iataS}-{iataE}";
}

public string GetIdAeAs()
{
    return getIdAeAs();
}

public Aeropuerto AeropuertoSalida
{
    get { return this._aeropuertoSalida; }
    set { this._aeropuertoSalida = value; }
}

public Aeropuerto AeropuertoEntrada
{
    get { return this._aeropuertoEntrada; }
    set { this._aeropuertoEntrada = value; }
}

public decimal Distancia { get { return this._distancia; } set { this._distancia = value; } }
}

public Ruta(Aeropuerto aeropuertoSalida, Aeropuerto aeropuertoEntrada, decimal
distancia)
{
    this._id = _idNext++;
    this._aeropuertoSalida= aeropuertoSalida;
    this._aeropuertoEntrada= aeropuertoEntrada;

```

```
        this._distancia= distancia;
    }

    public Ruta() { }

    private int getId()
    {
        return this._id;
    }

    public int GetId()
    {
        return getId();
    }

    private decimal getDistancia()
    {
        return this._distancia;
    }

    public decimal GetDistancia()
    {
        return getDistancia();
    }

    private Aeropuerto getAeropuertoE()
    {
        return this._aeropuertoEntrada;
    }

    public Aeropuerto GetAeropuertoE()
    {

```

```

        return getAeropuertoE();
    }

    private Aeropuerto getAeropuertoS()
    {
        return this._aeropuertoSalida;
    }

    public Aeropuerto GetAeropuertoS()
    {
        return getAeropuertoS();
    }

    public override string ToString()
    {
        return $"Id:{this._id}\nAeropuerto
salida:{this._aeropuertoSalida}\nAeropuerto
entrada:{this._aeropuertoEntrada}\nDistancia:{this._distancia}";
    }

}
}

```

```
//Premium
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Text;
```

```
namespace Dominio
```

```

{
    public class Premium : Cliente
    {
        //UML HECHO

        private int _puntos = 0;

        public int Puntos { get; set; }


        public Premium(string mail, string user, string password, int ci, string nombre,
string nacionalidad) : base( mail, user, password, ci, nombre, nacionalidad)
        {

        }

        public Premium() { }


        public override string ToString()
        {
            string baseInfo = base.ToString();

            return $"{baseInfo}\nTipo Cliente:Premium\nPuntos:{_puntos}";
        }

    }
}

//Pasaje
using System;

using System.Collections.Generic;

using System.Text;

namespace Dominio

```

```

{
    public class Premium : Cliente
    {
        //UML HECHO

        private int _puntos = 0;

        public int Puntos { get; set; }


        public Premium(string mail, string user, string password, int ci, string nombre,
string nacionalidad) : base( mail, user, password, ci, nombre, nacionalidad)
        {

        }

        public Premium() { }


        public override string ToString()
        {
            string baseInfo = base.ToString();

            return $"{baseInfo}\nTipo Cliente:Premium\nPuntos:{_puntos}";
        }

    }
}

//Ocacional
using System;

using System.Collections.Generic;

using System.Text;

namespace Dominio

```

```

{
    public class Ocacional : Cliente
    {
        //hecho UML

        private bool _elegible = false;


        public bool Elegible { get; set; }

        Random rnd = new Random();


        public Ocacional(string mail, string user, string password, int ci, string nombre,
string nacionalidad) : base(mail, user, password, ci, nombre, nacionalidad)
        {
            bool random = rnd.Next(0, 2) == 1;

            this._elegible = random;
        }

        public Ocacional() { }


        public override string ToString()
        {
            string baseInfo = base.ToString();

            string response = "";

            if (_elegible)
            {
                response = "Si";
            }
        }
    }
}

```



```

        else
        {
            response = "No";
        }

        return $"{baseInfo}\nTipo CLiente: Ocacional\nEs elegible? {response}";
    }
}

```

//Frecuencia Vuelo

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Text;
```

```
namespace Dominio
```

```
{
```

```
    //uml creado
```

```
    public enum FrecuenciaVuelo
```

```
    {
```

```
        Lunes,
```

```
        Martes,
```

```
        Miercoles,
```

```
        Jueves,
```

```
        Viernes,
```

```
        Sabado,
```

```
        Domingo
```

```
    }
```

```
}
```

```
//Equipaje
using System;
using System.Collections.Generic;
using System.Text;
```

```
namespace Dominio
{
    //UML CREADO
    public enum Equipaje
    {
        LIGHT,
        CABINA,
        BODEGA
    }
}
```

```
//CLIENTE
using System;
using System.Collections.Generic;
using System.Text;
```

```
namespace Dominio
{
    public class Cliente : Usuario
    {
        //uml hecho
        private int _ci;
```

```
private string _nombre;  
private string _nacionalidad;
```

```
public int Ci { get; set; }  
public string Nombre { get; set; }  
public string Nacionalidad { get; set; }
```

```
public Cliente(string mail, string user, string password,int ci, string nombre, string  
nacionalidad) : base(mail, user, password)
```

```
{  
    this._ci = ci;  
    this._nombre = nombre;  
    this._nacionalidad = nacionalidad;  
}
```

```
public Cliente() { }
```

```
private string getNombre()  
{  
    return this._nombre;  
}
```

```
public string GetNombre()  
{  
    return getNombre();  
}
```

```
private string getNacionalidad()
```

```

    {
        return this._nacionalidad;
    }

    public string GetNacionalidad()
    {
        return getNacionalidad();
    }

    private int getCi()
    {
        return this._ci;
    }

    public int GetCi()
    {
        return getCi();
    }

    public virtual string ToString()
    {
        return
        $"Nombre:{this._nombre}\nEmail:{this.Mail}\nNacionalidad:{this._nacionalidad}";
    }

}

}

//AVION
using System;

using System.Collections.Generic;

```

```
using System.Text;

namespace Dominio
{
    public class Avion
    {
        //UML hecho
        private static int _nextId = 1;

        private int _id;
        private string _fabricante;
        private string _modelo;
        private int _cantAsientos;
        private decimal _alcanceKm;
        private decimal _costoKm;

        public string Fabricante { get; set; }
        public string Modelo { get; set; }
        public int CantAsientos { get; set; }
        public decimal AlcanceKm { get; set; }
        public decimal CostoKm { get; set; }

        public Avion(string fabricante, string modelo, int cantAsientos, decimal
alcanceKm, decimal costoKm)
        {
            this._id = _nextId++;
            this._fabricante = fabricante;
            this._modelo = modelo;
```

```
        this._cantAsientos = cantAsientos;

        this._alcanceKm = alcanceKm;

        this._costoKm = costoKm;
    }

    public Avion() { }
```

```
    private string getModelo()
    {
        return this._modelo;
    }
```

```
    public string GetModelo()
    {
        return getModelo();
    }
```

```
    private decimal getCostoKm()
    {
        return this._costoKm;
    }
```

```
    public decimal GetCostoKm()
    {
        return getCostoKm();
    }
```

```
    private int getId()
    {
        return this._id;
    }
```

```
public int GetId()
```

```
{
```

```
    return getId();
```

```
}
```

```
private int getCantAsientos()
```

```
{
```

```
    return this._cantAsientos;
```

```
}
```

```
public int GetCantAsientos()
```

```
{
```

```
    return getCantAsientos();
```

```
}
```

```
private string getIFC()
```

```
{
```

```
    return $"ID:{this._id} | Fabricante: {this._fabricante} | Modelo: {this._modelo}";
```

```
}
```

```
public string GetIFC()
```

```
{
```

```
    return getIFC();
```

```
}
```

```
public override string ToString()
```

```
{
```

```
    return $"ID:{this._id}\nEl fabricante es: {this._fabricante}\nEl modelo es:  
{this._modelo}\nLa cantidad de asientos es: {this._cantAsientos}\nEl Alcance(km) es:  
{this._alcanceKm}\nEl costo(km) es: {this._costoKm}\n";
```

```

    }

}

}

//AEROPUERTO
using System;

using System.Collections.Generic;

using System.Text;

using System.Threading.Tasks;

namespace Dominio
{
    public class Aeropuerto
    {
        //UML HECHO
        private static int _nextId = 1;

        private int _id;

        private string _iata;

        private string _ciudad;

        private decimal _costoOperacion;

        private decimal _costoTasa;


        public string Iata { get { return _iata; } set { _iata = value; } }

        public string Ciudad { get { return _ciudad; } set { _ciudad = value; } }

        public decimal CostoOperacion { get { return _costoOperacion; } set {
            _costoOperacion = value; } }

        public decimal CostoTasa { get { return _costoTasa; } set { _costoTasa = value; } }
    }
}

```



```
public Aeropuerto(string iata,string ciudad,decimal costoOperacion,decimal
costoTasa)
```

```
{
    this._id = _nextId++;
    this._iata = iata;
    this._ciudad = ciudad;
    this._costoOperacion = costoOperacion;
    this._costoTasa = costoTasa;
}
```

```
public Aeropuerto() { }
```

```
public override string ToString()
```

```
{
    return $"Codigo IATA:{this._iata}\nCiudad:{this._ciudad}\nCosto
Operacion:{this._costoOperacion}\nCosto de tasa:{this._costoTasa}\n";
}
```

```
private string getIata()
```

```
{
    return this._iata;
}
```

```
public string GetIata()
```

```
{
    return getIata();
}
```

```
private int getId()
```

```
{
```

```
        return this._id;
    }

    public int GetId()
    {
        return getId();
    }

    private decimal getCostoOperacion()
    {
        return this._costoOperacion;
    }

    public decimal GetCostoOperacion()
    {
        return getCostoOperacion();
    }

    private decimal getCostoTasa()
    {
        return this._costoTasa;
    }

    public decimal GetCostoTasa()
    {
        return getCostoTasa();
    }

    private string getCodigo()
    {
        return this._iata;
    }

    public string GetCodigo()
```

```

    {
        return getCodigo();
    }
}

```

```
//ADMINISTRADOR
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Text;
```

```
namespace Dominio
```

```

{
    public class Administrador : Usuario

```

```

    {
        //hecho uml
        private string _apodo;

```

```

        public string Apodo {
            get { return _apodo; }
            set { _apodo = value; }
        }

```

```

        public Administrador(string mail,string user,string password,string apodo) :
        base(mail,user,password)

```

```

        {
            this._apodo = apodo;
        }

```

```
        public Administrador() { }
```

```
private string getApodo()
{
    return this._apodo;
}

public string GetApodo()
{
    return getApodo();
}

public override string ToString()
{
    return $"Usuario:{this.User}\nEmail:{this.Mail}\nApodo:{this._apodo}\n";
}
}
```