



Alumno: _____ DNI: _____

RuynAir.

Duración: 3h

**Problema. Análisis de Viabilidad (20% nota).**

La empresa Ruynair, que actualmente lleva la **facturación** mediante sistema cliente/servidor está pensando en implantar un sistema informático basado en "Computación en la nube" para un proceso más rápido y fiable. Hasta ahora disponía de un ordenador servidor (renting renovable anualmente, a razón de 1100€/año durante próximos 3 años) con un contrato de mantenimiento anual de 350€/año, una persona que dedica 65 horas/mes a su gestión (sueldo global 1700€/mes). El nuevo sistema requeriría un *upgrade* de la línea de datos (80€/mes) y un coste del proveedor del servicio de 170€ / año. La misma persona dedicada a gestión habrá de dedicar un 20% más de tiempo, con una formación del nuevo sistema de 20 horas.

1. ¿Es capaz de cuantificar el beneficio/coste en cada año?, (1.5 puntos)

ingresos	año 1	año 2	año 3
Renting	1100	1100	1100
mantenimiento	350	350	350
gestion	650	650	650
	0	0	0
Total ingresos	2100	2100	2100
Gastos			
gestion	780	780	780
upgrade	960	960	960
proveedor	170	170	170
formacion	200	0	0
Total gastos	2110	1910	1910
Beneficio neto anual	-10	190	190
Beneficio acumulado	-10	180	370

2. ¿En qué mes y año se produce el punto de equilibrio, en que se igualan los gastos ocasionados con los beneficios conseguidos? (0.5 puntos)

NOTA: Mes con 170 horas laborables

 $t \cdot (190/12) = 10$; $t = 0,63$ meses del 2º año, mediados de enero.

Visión del negocio

Una vez externalizada la gestión de la facturación, debemos analizar y modelar el sistema de gestión de reserva de vuelos de RyanAir. El sistema debe permitir al usuario hacer consultas y reservas de vuelos, además de poder comprar los billetes aéreos de forma remota a través de internet, sin la necesidad de recurrir a un agente de viajes humano.

Cuando se entra en el sistema se presenta un mensaje de bienvenida describiendo los servicios ofrecidos junto con la opción para **registrarse** por primera vez, o si ya se está registrado, poder utilizar el sistema de reserva de vuelos. Este acceso se da por medio de la inserción de un login previamente especificado (dirección de correo electrónico del usuario) y una contraseña previamente escogida y que debe **validarse**.

Una vez registrado el usuario, y después de haberse validado el registro y contraseña del usuario, se pueden seleccionar las siguientes actividades:

- **Consulta** de vuelos.
- **Reserva** de vuelos.
- **Compra** de billetes.
- **Insultar** a Michael O'Leary

La consulta de vuelos se puede hacer de tres maneras diferentes:

- **Horarios** de Vuelos.
- **Tarifas** de Vuelos.
- **Información** de Vuelo

La consulta *según horario* muestra los horarios de las diferentes aerolíneas que dan servicio entre dos ciudades. La consulta *según tarifas* muestra los diferentes vuelos entre dos ciudades ordenados por su costo.

La *información* de vuelos se utiliza principalmente para consultar el estado de algún vuelo, incluyendo información de si existen asientos disponibles y, en el caso de un vuelo para el mismo día, si éste está en hora.

La *reserva* de vuelo permite al cliente hacer una reserva para un vuelo particular, especificando la fecha y horario, bajo una tarifa establecida.

La *compra* permite al cliente, dada una reserva de vuelo previa y una tarjeta de crédito válida, adquirir los billetes aéreos. Es necesario estar previamente registrado con un número de tarjeta de crédito válida para poder hacer compras de billetes, o bien proveerla en el momento de la compra.

Como casi seguro que su viaje será decepcionante, le damos la posibilidad de *insultar* a nuestro Consejero Delegado, pudiendo incluso incluir una foto o audio ofensivo. *No le servirá de mucho pero se quedará a gusto.*

La aerolínea posee una dotación de aviones, de distintas marcas, modelos y capacidad. La aerolínea asigna a cada avión la realización de una serie de vuelos. La Aerolínea provee servicio de múltiples vuelos entre diferentes ciudades bajo diferentes horarios. La aerolínea se identifica por un nombre.

El Vuelo se identifica por medio de un número, tiene como origen un aeropuerto en una ciudad y tiene como destino un aeropuerto de otra ciudad. Así mismo, el vuelo puede operar varios días a la semana teniendo un horario de salida y otro de llegada. Un vuelo puede tener múltiples escalas (paradas intermedias, antes de llegar a destino). Los vuelos se relacionan por medio de conexiones (dos vuelos se conectan si uno tiene como origen el destino de otro con tiempo suficiente para facturar el equipaje).

El aeropuerto se encuentra en una ciudad de un país determinado. El Aeropuerto sirve como origen, destino o escalas de un vuelo.

Un vuelo en una fecha determinada se hace en un tipo de avión particular. El tipo de avión define la cantidad máxima de pasajeros que pueden viajar en ese vuelo para esa fecha.

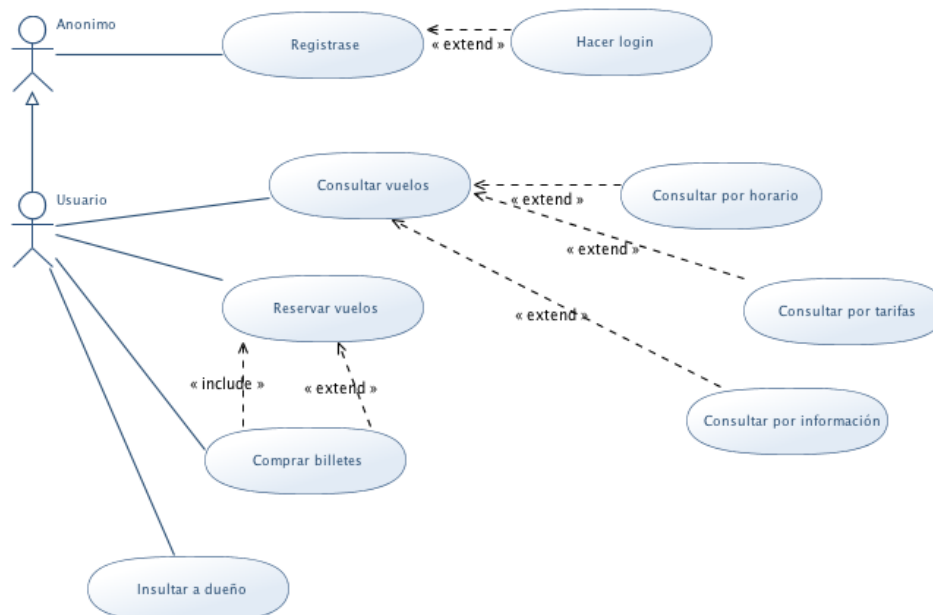
Los diferentes vuelos tienen múltiples tarifas para compra de billete, variando según la clase de billete, si son de ida o de ida y vuelta, y dependiendo de las diversas restricciones y ofertas existentes.

Problema. Modelado (40% nota).

Se le pide que analice, diseñe e implemente los siguientes entregables:

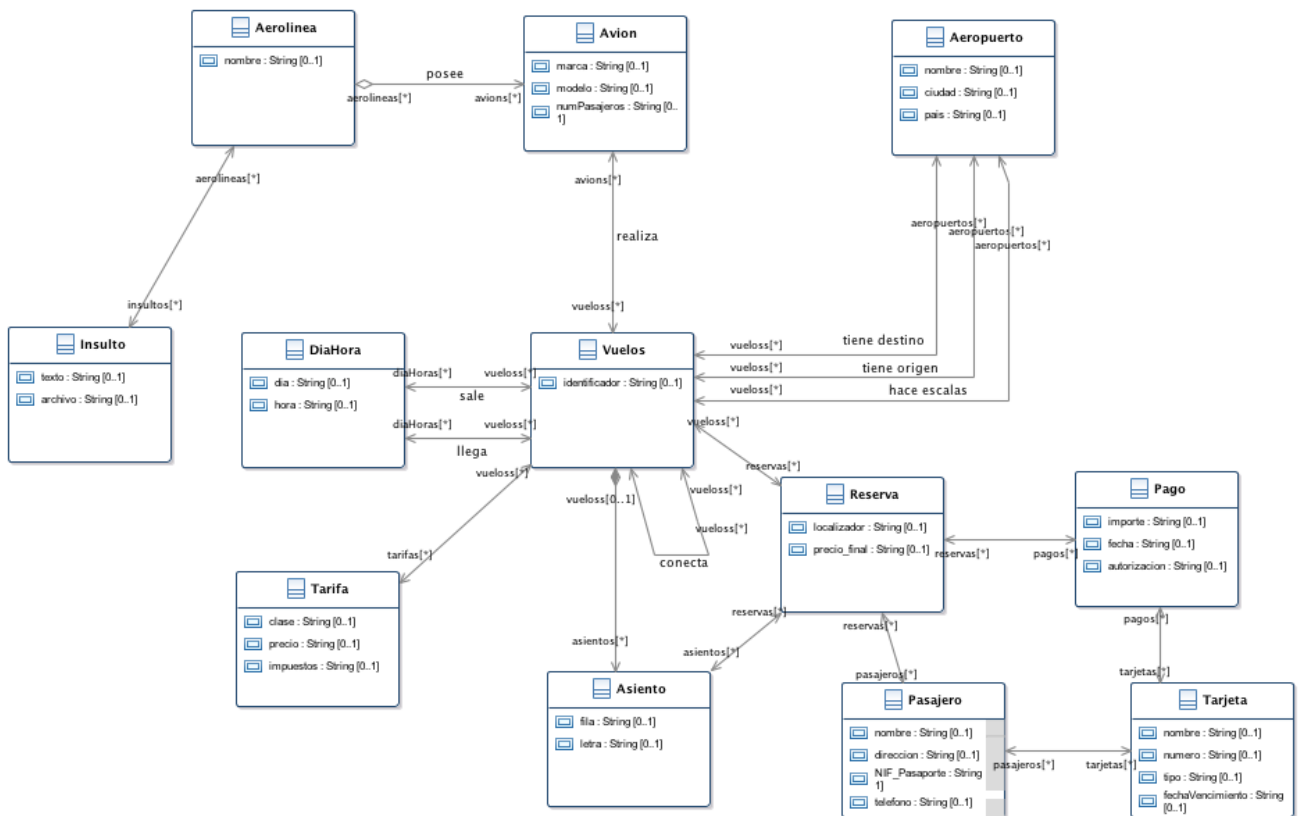
3. Diagrama de Casos de Uso

(2 puntos)



4. Diagrama de Clases

(2 puntos)



Problema. Construcción (12.5% nota).

Una vez terminados el análisis y modelado, y realizada la estimación del proyecto, empezamos a codificar. Para construir el software, uno de los primeros pasos será, a partir del **Diagrama de Clases** obtenido, implementar en Java la estructura de las clases.

5. Realice la implementación estructural en Java de las clases **Vuelo** y **Reserva** (incluya sólo la declaración de los atributos, ni incluya ningún método, ni los *getters* y *setters* de los atributos).

(1.25 puntos)

```
public class Vuelo{
    private Long identificador;
    private List<DiaHora> salidas;
    private List<DiaHora> llegadas;
    private List<Tarifa> precios;
    private List<Asiento> mAsientos;
    private List<Vuelos> conectaConVuelo;
    private List<Reservas> mReservas;
    private Avion realizaVuelo;
    private Aeropuerto aeropuertoOrigen;
    private Aeropuerto aeropuertoDestino;
    private List<Aeropuerto> aeropuertosEscalas;
}
```

```
public class Pasajero{
    private Long localizador;
    private Long precioFinal;
    private Asiento asientoReservado;
    private Vuelo vueloReservado;
    private Pago pagoRealizado;
    private Pasajero titularReserva;
}
```

Problema. Pruebas. (27.5% nota) Dado el siguiente programa en C:

```
# include <stdio.h>
# include <conio.h>
int main () {
    int a, b, c, boolean = 0;
    printf ("\n\t Introduzca lado a:");
    scanf ("%d", &a);
    printf ("\n\t Introduzca lado b:");
    scanf ("%d", &b);
    printf ("\n\t Introduzca lado c:");
    scanf ("%d", &c);
    if ((a > 0) && (a < 100) && (b > 0) && (b < 100) && (c > 0) && (c < 100))
    {
        if ((a + b) > c) && ((c + a) > b) && ((b + c) > a))
        {
            boolean = 1;
        }
    }
    else {
        boolean = -1;
    }
    if (boolean == 1) {
        if ((a == b) && (b == c)) {
            printf ("Triangulo es equilatero");
        }
        else if ((a == b) || (b == c) || (c == a))
        {
            print ("Triangulo es isosceles");
        }
        else {
            printf("Triangulo es escaleno");
        }
    }
}
```

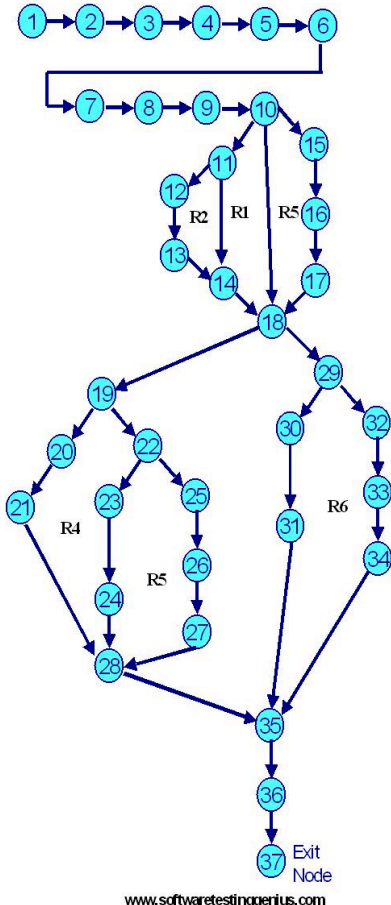
```

    }
    else if (boolean == 0) {
        printf ("No es un triangulo");
    }
    else
        printf ("\n rango de valores no valido");
}
getch();
return -1;

```

6. Calcule de dos maneras distintas su Complejidad Ciclomática.

(1,5 puntos)



```

(1) int main ( )
(2) {
(3) int a, b, c, boolean = 0;
(4) printf ("nt Enter side-a :");
(5) scanf ("%d", & a);
(6) printf("nt Enter side-b :");
(7) scanf ("%d", & b);
(8) printf ("nt Enter side-c:");
(9) scanf ("%d", & c);
(10) if ((a > 0) && (a < - 100) && (b > 0) && (b < . 100) && (c > 0) && (c <
=100)) {
(11) if ((a + b) > c) && ((c + a) > b) && ((b + c) > a)) {
(12) boolean = 1;
(13) }
(14) }
(15) else {
(16) boolean = -1;
(17) }
(18) if (boolean == 1) {
(19) if ((a == b) && (b == c)) {
(20) printf ("Triangle is equilateral");
(21) }
(22) else if ((a == b) || (b == c) || (c == a)) {
(23) print ("Triangle is isosceles");
(24) }
(25) else {
(26) printf("Triangle is scalene");
(27) }
(28) }
(29) else if (boolean == 0) {
(30) printf ("Not a triangle");
(31) }
(32) else
(33) printf ("n invalid input range");
(34) }
(35) getch ( );
(36) return -1;
(37) }

```

$$V(G)=A-N+2=23-18+2=7$$

$$V(G)=C+1=6+1=7$$

$$V(G)=R+1=6+1=7$$

7. La Complejidad Ciclomática se suele usar en la fase de pruebas con una determinada finalidad.

¿Cuál es? Aplíquela.

(0,75 puntos)

Analiza la estructura de control del software construyendo un grafo de flujo que abstraer el código y Mide el número de caminos linealmente independientes de un programa o, dicho de otro modo, define el número de caminos mínimos que recorran todos las sentencias (cobertura de las pruebas).

Los 7 caminos podrían ser:

1-2-3-4-5-6-7-8-9-10-15-16-17-18-29-32-33-34-35-36-37

1...10-18-29-32-33-34-35-36-37

1...10-11-14-18-29-32-33-34-35-36-37

1...10-11-12-13-14-18-19-22-23-24-28-35-36-37

...

8. La Complejidad Ciclomática es una métrica que nos puede ser muy útil en la fase de mantenimiento.

¿Por qué?

(0,5 puntos)

También se usa como medida de la complejidad, a mayor valor de $V(G)$, mayor complejidad del programa. Diversos estudios han demostrado una fuerte correlación entre la métrica y los errores existentes en código. Por ello, código con alta $V(G)$ serán propensos a errores y habrá que tener una especial precaución con ellos.