
Ejercicios del Tema 1: Construcción y Pruebas del Software

Ejercicio 1

Se desean realizar pruebas de la caja negra sobre un programa utilizado por una empresa de transporte para calcular la tarifa de cada billete según el trayecto, la antelación en la que se obtiene el billete y la edad del pasajero. Dicha empresa sólo opera viajes entre Santander, Madrid y Barcelona.

Como datos de entrada toma:

- **CiudadOrigen** que es un campo que puede tomar los valores “SNT”, “MAD” y “BCN”.
- **CiudadDestino** que puede tomar los mismos valores “SNT”, “MAD” y “BCN”.
- **Fecha** es un campo del tipo fecha que indica el día en el que se pretende realizar el viaje.
- **Edad** es un campo numérico positivo de 3 cifras (incluyendo el 000).

La tarifa obtenida además de estar en función del trayecto realizado, ofrece los siguientes descuentos por antelación y edad del pasajero. Los descuentos no son acumulables y siempre se aplicará el de mayor valor.

- **15%** de descuento sacando el billete con antelación superior a 1 semana y **25%** con antelación superior a 1 mes.
- **30%** a los pasajeros con edad inferior a 25 años y **40%** a los pasajeros con edad superior a 65 años.

Se pide:

- 1.1.** Realizar una tabla con las clases de equivalencia indicando las clases válidas y no válidas para cada variable de entrada.
- 1.2.** Obtener casos de prueba de dicha tabla, indicando las clases de equivalencia que cubriría cada caso (numerar previamente las clases).
- 1.3.** Aplicar la técnica de análisis de valores límite para obtener más casos de prueba que pudieran presentar un tratamiento diferenciado.

Ejercicio 2

Dado el siguiente pseudocódigo

```
Do {  
    Puntero=falso;  
    If (x>k)  
        Then Puntero=verdad;  
        x=x+1;  
        Call sub(x, puntero, resultado); }  
While(resultado>0);  
Imprimir(resultado);
```

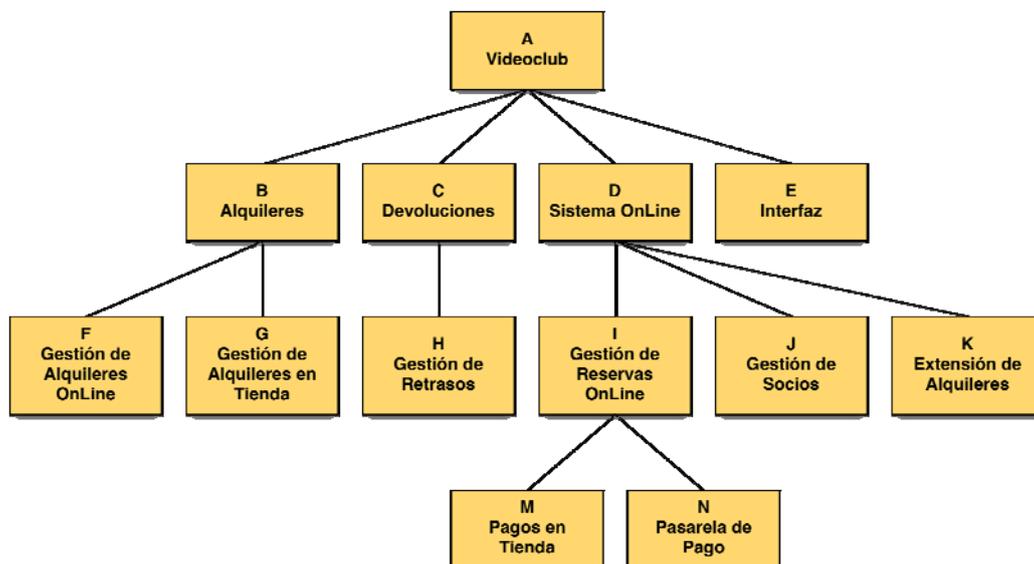
Se pide:

- Obtener el grafo de flujo del programa señalando los trozos de código que corresponden con cada nodo.
- Calcular la complejidad ciclomática mediante las tres fórmulas, señalando las regiones sobre el grafo.
- Definir 3 conjuntos de caminos (indicando la secuencia de nodos que atraviesa cada camino) necesarios para probar todas las sentencias, ramas y caminos.

Ejercicio 3

La siguiente jerarquía de componentes describe algunos módulos de un sistema software para un videoclub. Indique cuál sería la secuencia de pruebas para la integración de componentes utilizando los siguientes enfoques:

- Integración incremental ascendente (bottom-up).
- Integración incremental descendente (top-down).
- Integración según hilos de funcionalidad.



Ejercicio 4

Un programa toma como entrada un fichero cuyo formato de registro es el siguiente:

- **Número-empleado** es un campo de números enteros positivos de 3 dígitos (excluido el 000).
- **Nombre-empleado** es un campo alfanumérico de 10 caracteres.
- **Meses-Trabajo** es un campo que indica el número de meses que lleva trabajando el empleado; es un entero positivo (incluye el 000) de 3 dígitos.
- **Directivo** es un campo de un solo carácter que puede ser «+» para indicar que el empleado es un directivo y «-» para indicar que no lo es.

| | | | |
|-----------------|-----------------|---------------|-----------|
| Numero-empleado | Nombre-empleado | Meses-Trabajo | Directivo |
|-----------------|-----------------|---------------|-----------|

El programa asigna una prima (que se imprime en un listado) a cada empleado según las normas siguientes:

- **P1** a los directivos con, al menos, 12 meses de antigüedad.
- **P2** a los no directivos con, al menos, 12 meses de antigüedad.
- **P3** a los directivos sin un mínimo de 12 meses de antigüedad.
- **P4** a los no directivos sin un mínimo de 12 meses de antigüedad.

Se pide:

4.1. Crear una tabla de clases de equivalencia (las clases deberán ser numeradas) en la que se indiquen las siguientes columnas en cada fila:

- Condición de entrada que se analiza.
- Clases válidas.
- Clases no válidas que se generan para la condición.

4.2. Partiendo de las clases de equivalencia creadas generar los casos de prueba:

- Enunciar la regla que se ha aplicado para derivar los casos a partir de las clases de equivalencia.
- Indicar las clases que cubre cada caso de prueba.
- Especificar la entrada en todos los casos y la salida esperada sólo en los casos válidos.

Ejercicio 5

El siguiente pseudocódigo cumple la especificación del problema anterior. Se han añadido algunas líneas de cabecera y de final al listado de las primas, y se han asignado valores numéricos concretos a las primas P1, P2, P3 y P4. El fichero de entrada se denomina FICH y los nombres de los campos son más o menos iguales al ejercicio anterior.

```
1 Begin {programa}
2   Print ("Ilmo.Sr.Director General:");
3   Read (registro_FICH);
4   Prima = 0;
5   While (no FF de FICH) do
6     Begin {while}
7       If (meses_FICH>=12)
8         Then
9           If (directivo_FICH="+")
10            then Prima=1000;
11            else Prima=75;
12          Else
13            If (directivo_FICH="+")
14              then Prima=500;
15              else;
16          Print (num_FICH, nombre_FICH, Prima);
17          Read (registro_FICH);
18        End {while};
19   Print ("S.e.u.o.");
20 End {programa}
```

Suponemos que en la anterior prueba de clases de equivalencia el programa se ejecuta sólo con los dos registros siguientes de casos válidos, consecutivamente y en el orden dado (el símbolo # indica un carácter blanco):

- Registro 1: 123 Fernández# 009 +
- Registro 2: 456 Fernando## 013 -

Se pide:

5.1 Comprobar si se cumple la cobertura de sentencias indicando, en su caso, cuáles de ellas no se ejecutan. En caso de que no se cumpla, añadir el mínimo número de registros adicionales para que se cumpla la cobertura.

5.2 Comprobar si se cumple la cobertura de decisiones, creando una tabla donde se marquen los valores que adopta cada decisión. En caso de no cumplirse la cobertura, añadir el mínimo número de registros adicionales para que se cumpla. Los casos a añadir deben definir la entrada y la salida esperada al ejecutarse junto a los dos casos de caja negra (registros 1 y 2). Emplear el formato de salida que se incluye en el código del programa.

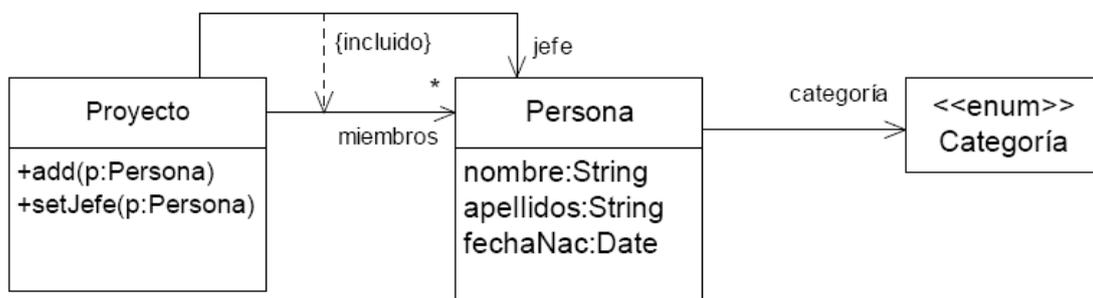
Ejercicio 6:

Escriba una clase para calcular el valor medio de 3 números que se pasan como parámetros. A continuación someta el programa al proceso de pruebas descrito en clase hasta que:

- Se hayan recorrido todas las sentencias.
- Se hayan recorrido todas las condiciones a true y a false.

Ejercicio 7:

Suponga que disponemos de un sistema como el mostrado en la figura. Proponga valores interesantes de tipo Persona para las operaciones add y setJefe del tipo Proyecto.



Ejercicio 8:

Escriba casos de prueba que cumplan los criterios de cobertura de transición de estados para la siguiente máquina de estados, que representa el comportamiento de una cuenta corriente.

