Para las pruebas de caja blanca, existe software específico para la realización de pruebas. Estas pruebas han de probar como mínimo 1 vez cada sentencia.

Los manuales han de ser correctos, es decir, si pone cómo instalar el software correctamente, siguiendo esos pasos ha de satisfacerse dicha instalación.

Pruebas de unidad o unitarias (sobre el código) → Pruebas de integración → Diseño → Pruebas de validación → Requisitos → Pruebas de sistema → Ingeniería de sistema

**Pruebas de unidad o unitarias**

Normalmente se realizan con un software.

**Pruebas de integración**

Comprobar cómo se relacionan las distintas partes del código, ya que pueden ser desarrolladas por distintas entidades.

Existen 2 tipos:

* No incrementales
* Incrementales:
  + Ascendentes
  + Descendentes

**Pruebas de validación**

Pruebas muy dirigidas, es decir, se comprueba que el software cumple con lo requerido. Existen las pruebas Alpha y Beta. Las primeras se realizan en un entorno más controlado.

**Pruebas de sistema**

Hay varios tipos:

* **De recuperación:** Comprobar si el software al fallar es capaz de recuperarse.
* **De seguridad:** Comprobar que el software es seguro.
* **De estrés:** Se somete al software con mucha carga de trabajo.

**Pruebas de funcionalidad**

* Caja blanca
  + Grafos→ Complejidad ciclomática de un programa
  + Casos de uso resultantes de grafos
* Caja negra
  + Análisis:
    - Particiones de equivalencia (Clases o rangos válidos y no válidos). Se usarán los valores resultantes en los distintos casos de uso.

**COMPLEJIDAD CICLOMÁTICA DE UN PROGRAMA**

Es una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa.

Contiene los caminos independientes y dependientes. Los primeros serían la complejidad ciclomática y que identifican los casos de uso mínimos, con los segundos se comprueban todos los casos.

**Clases de complejidad**

De entre 1 a 10 bajo

De entre 11 a 20 moderado

De 21 a 50 alto riesgo

Mayor que 50 no pueden ser testeables

Se realiza un diagrama de grafo.

El cálculo de esta se realiza de distintos métodos:

1. Número de regiones
2. Número de aristas – número de nodos + 2
3. Número de nodos predicado + 1

**Regiones**: Zonas cerradas y rodeadas de nodos.

**Aristas**: Flujo que sigue el programa (las flechas)

**Nodo normal:** sentencias normales

**Nodo predicado**: recibe varias entradas o varias salidas

**Ejercicios en cuaderno**

**CASOS DE USO SEGÚN DIAGRAMAS DE GRAFO**

Ejercicio de pseudocódigo del cuaderno. Casos de prueba:

1. 1-6-8: estudiantes > 10 / aprobados =< 8
   1. Estudiantes = 11
   2. Aprobados = 8
2. 1-6-7-8: estudiantes > 10 / aprobados > 8
   1. Estudiantes = 11
   2. Aprobados = 9
3. 1-2-4-5-1-6-8: estudiantes <= 10 / resultado = 1 / aprobados <= 8
   1. Estudiantes = 9
   2. Resultado = 1
   3. Aprobados = 8
4. 1-2-3-5-1-6-8: estudiantes <= 10 / resultado ¡= 1 / aprobados <= 8
   1. Estudiantes = 9
   2. Resultado = 2
   3. Aprobados = 8
5. 1-2-4-5-1-6-7-8: estudiantes <= 10 / resultado ¡= 1 / aprobados > 8
   1. Estudiantes = 9
   2. Resultado = 2
   3. Aprobados = 9
6. 1-2-3-5-1-6-7-8: estudiantes <= 10 / resultado = 1 / aprobados > 8
   1. Estudiantes = 9
   2. Resultado = 1
   3. Aprobados = 9

**RANGOS VÁLIDOS DE DATOS**

Existe un rango válido y dos NO válidos, los que están por debajo y por encima de dicho rango válido.

**Nota:**

* Rango válido: {1, 10}
* Rangos no válidos: {-Infinito, 0} / {11, +Infinito}

Es importante comprobar también los casos con los rangos no válidos.

**PRUEBAS DE CAJA NEGRA**

**Salario según trabajo:**

* Nº emp→ Rango válido 0-100
* Puesto→ Conjunto {Cajero, Reponedor, Gerente}
* Salario→ Rango (Salario mínimo – Máxima salario según categoría)

Con estos datos se hace una tabla:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONDICIONES DE ENTRADA | CLASES DE EQUIVALENCIA | CLASES VÁLIDAS | CÓDIGO | CLASES NO VÁLIDAS | CÓDIGO |
| Nº EMPLEADO | RANGO | 1 – 100 | V1.1 | <1 | F1.1 |
| >100 | F1.2 |
| CONJUNTO | ENTERO | DECIMAL | F1.3 |
| PUESTO | CONJUNTO | CAJERO | V2.1 | PUESTO NO VÁLIDO | F2.1 |
| REPONEDOR | V2.2 |
| GERENTE | V2.3 |
| SALARIO | RANGO | 1000-1500 | V3.1 | <1000 | F3.1 |
| >1500 | F3.2 |
| 1000-1200 | V3.2 | <1000 | F3.3 |
| >1200 | F3.4 |
| 1000-2000 | V3.3 | <1000 | F3.5 |
| >2000 | F3.6 |

A partir de esta tabla, se generará una nueva tabla con los casos de prueba posibles:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CASO DE PRUEBA | CLASES DE EQUIVALENCIA | Nº EMPLEADO | PUESTO | SALARIO | RESULTADO ESPERADO |
| CP1 | V1.1, V2.1, V3.1 | 50 | CAJERO | 1200 | SE ALMACENA E INFORMA |
| CP2 | V1.1, V2.2, V3.2 | 75 | REPONEDOR | 1100 | SE ALMACENA E INFORMA |
| CP3 | V1.1, V2.3, V3.3 | 25 | GERENTE | 1500 | SE ALMACENA E INFORMA |
| CP4 | V1.1, F2.1, V3.1 | 50 | SEGURIDAD | 1200 | SE INDICA QUE EL PUESTO NO ES VÁLIDO |
| CP5 | V1.1, V2.1, F3.1 | 1 | CAJERO | 500 | SE INDICA QUE EL SALARIO NO ES VÁLIDO PARA LA CATEGORÍA CAJERO |
| CP6 | V1.1, V2.1, F3.2 | 100 | CAJERO | 1750 | SE INDICA QUE EL SALARIO NO ES VÁLIDO PARA LA CATEGORÍA CAJERO |
| CP7 | V1.1, V2.2, F3.3 |  |  |  |  |
| CP8 | V1.1, V2.2, F3.4 |  |  |  |  |
| CP9 | V1.1, V2.3, F3.5 |  |  |  |  |
| CP10 | V1.1, V2.3, F3.6 |  |  |  |  |
|  | F1.1, V2.1, V3.1 |  |  |  |  |
|  | F1.1, V2.2, V3.2 |  |  |  |  |
|  | F1.1, V2.3, V3.3 |  |  |  |  |
|  | F1.1, F2.1, V3.1 |  |  |  |  |
| … | … | … | … | … | … |

**ANÁLISIS EXTREMAL**

Se comprueban los valores límites que se admiten y valores no válidos alejados de estos.