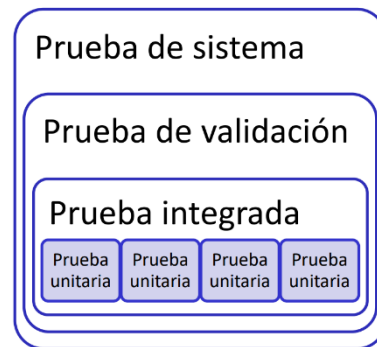


# Comprueba tu aprendizaje 1 (pag 168)

1.- ¿Qué estrategias se siguen para probar el software? Si las pruebas de unidad funcionan, ¿es necesario hacer la prueba de integración?

1. Prueba unitaria o de unidad
2. Prueba de integración
3. Prueba de validación o de aceptación
4. Prueba de sistema



Prueba unitaria	<ul style="list-style-type: none"><li>• se centra en la unidad más pequeña del SW, el módulo tal como está implementado en código fuente</li></ul>
Prueba integrada	<ul style="list-style-type: none"><li>• se toman módulos probados en la P.Unitaria y se construye una estructura de programa que esté de acuerdo con lo que dicta el diseño</li><li>• foco de atención: el diseño</li></ul>
Prueba de aceptación	<ul style="list-style-type: none"><li>• Prueba del SW en el entorno real de trabajo (producción) o en un entorno de pre-producción (copia del real) con intervención del usuario final.</li><li>• Se validan los requisitos establecidos como parte del análisis de requisitos del SW comparándolos con el sistema que ha sido construido.</li></ul>
Prueba de sistema	<ul style="list-style-type: none"><li>• Verifica que cada elemento encaja de forma adecuada y se alcanza la funcionalidad y rendimiento total.</li><li>• Se prueba como un todo el software y otros elementos del sistema.</li></ul>

## ¿es necesario hacer la prueba de integración?

Si, ya que el de integración sirve para observar como interaccionan los distintos módulos, es decir, para comprobar si funcionan juntos. Y aunque en uno de los enfoques se haga primero por separado y luego se junten, el de unidad solo prueba los módulos individualmente y más como internamente que la integración, sin el de integración no podríamos comprobar el funcionamiento en conjunto de los módulos.

# Comprueba tu aprendizaje 5 (pag 170)

5.- Realiza el grafo de flujo, calcula la complejidad ciclomática, define el conjunto básico de caminos, elabora los casos de prueba para cada camino y evalúa el riesgo para la siguiente función Java:

```
static int Contador1(int x, int y) {
```

```
1) Scanner entrada = new Scanner(System.in);
```

```
2) int num, c = 0;
```

```
3) if (x > 0 && y > 0) {
```

```
4)   System.out.println("Escribe un número");
```

```
5)   num = entrada.nextInt();
```

```
6)   if (num >= x && num <= y){
```

```
7)     System.out.println("\tNúmero en el rango");
```

```
8)     c++;
```

```
   }
```

```
   else
```

```
9)   System.out.println("\tNúmero fuera de rango");
```

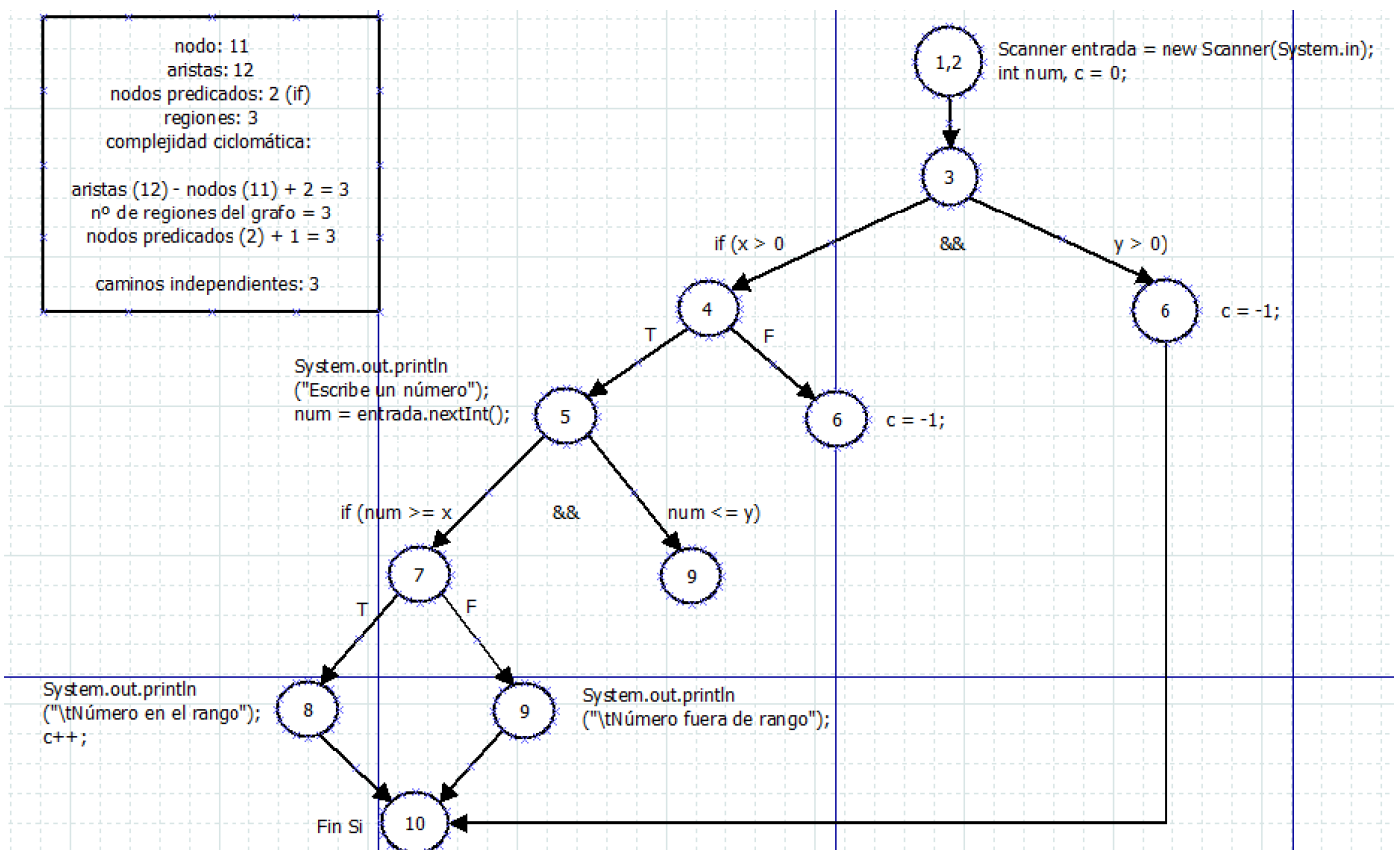
```
   }
```

```
   else
```

```
10) c = -1;
```

```
11) return c;
```

```
//
```



Condición de entrada	Clases de equivalencia	Clases válidas	COD	Clases no válidas	COD
Variable “x”	Valor	Nº > = 0	V1	Nº < 0	NV1
Variable “y”	Valor	Nº > = 0	V2	Nº < 0	NV2
Número de Cuenta	Valor	Nº de 5 dígitos	V4	Nº de menos de 5 dígitos Nº de más de 5 dígitos	NV6 NV7
Clave personal	Valor	Alfanumérico de 5	V5	Alfanumerico de menos de 5 po. Alfanumerico de más de 5 po.	NV8 NV9
Orden	Lógica	En blanco	V6	No es un string	NV10
	Valor	“Talónario” “movimiento”	V7	Palabra diferente a una de éstas	NV11

Si la entrada es correcta, el programa otorga un documento a cada cliente según estas normas:

- S1: Si Orden tiene el valor “Talónario”, el usuario recibe un talonario de cheques.
- S2: Si Orden tiene el valor “Movimientos”, el usuario recibe los movimientos del mes en curso.
- S3: Si Orden está en blanco, el usuario recibe los dos documentos.

Si la entrada no es correcta, el programa muestra un mensaje de error sobre el dato implicado:

- ER1: Si “Talónario” no es correcto.
- ER2: Si “Movimientos” no es correcto.

Caso de Prueba	Clases de Equivalencia	Condiciones de Entrada			Resultado esperado
		“Talónario”	“Movimientos”	En blanco	
CP1	V7	Si	-	-	S1
CP2	V7	-	Si	-	S2
CP3	V6	-	-	Si	S3
CP4	NV10, NV11	Si	Si	-	ER1

Condiciones de entrada y de salida	Casos de prueba
Una variable toma valores comprendidos entre -4 y 4 (enteros)	Valores: -4, -3, 3, 4
El programa lee un fichero que contiene de 1 a 100 registros	Leer el fichero 0, 1, 100, 101 registros
El programa deberá generar de 1 a 5 listados	Generar de 0, 1, 5 y 6 listados
El número de alumnos para calcular la nota media es 35	Para 0, 1, 35, 36
La función deberá devolver un array de enteros, de 1 a 10 valores.	Devolver array con 0, 1, 10, y 11 valores