

CFGS

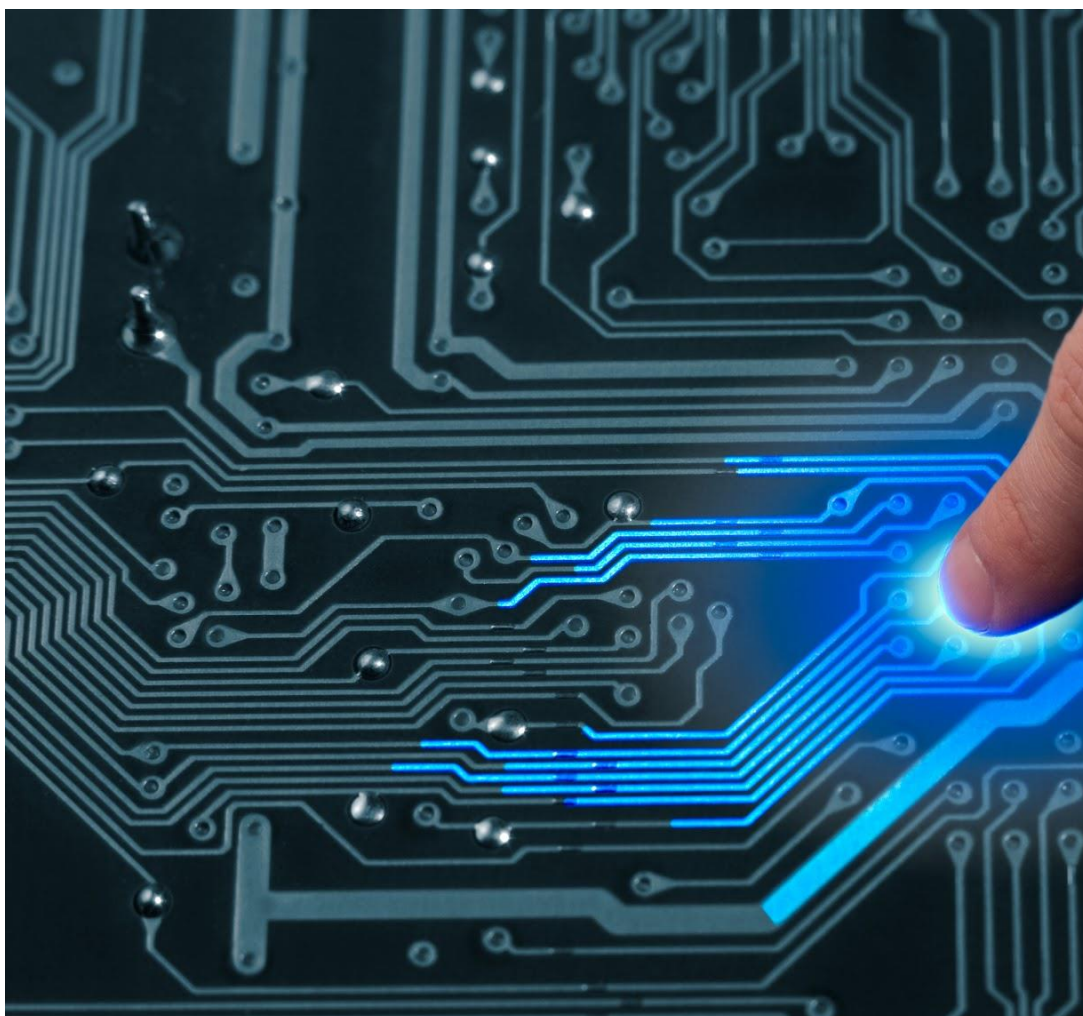
DESARROLLO DE APLICACIONES MULTIPLATAFORMA

DESARROLLO DE APLICACIONES WEB

ENTORNOS DE DESARROLLO

OPTIMIZACIÓN DE SOFTWARE

PAC 1: Diseño y realización de pruebas



PAC 1: Diseño y realización de pruebas

INFORMACIÓN

Para responder a las siguientes cuestiones deberás ayudarte del material didáctico y consultar internet.

Requisitos varios que deben cumplirse en vuestros trabajos:

- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- Siempre que utilicéis información del material didáctico para responder / resolver alguna pregunta, tenéis que citar el tema y la página de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- Se valorará la presentación, ortografía y gramática de vuestro trabajo hasta con un punto de la nota final.

1. ¿Qué diferencias existen entre pruebas de caja blanca y pruebas de caja negra?

Las pruebas de caja blanca o estructurales se centran en validar los requisitos funcionales especificados, comprobando la salida que muestra respecto a las diferentes entradas de datos posibles. Se encarga de ejecutar todos los caminos independientes que se encuentran en el programa.

Las pruebas de caja negra o de comportamiento se centran en el proceso, es decir, en el funcionamiento interno del programa. Su función es encontrar los diferentes errores que pueden surgir.

2. Comenta alguna característica de cada una de las estrategias de pruebas.

La estrategia de pruebas de software se compone de un ciclo de cuatro tipos de pruebas: unidad, integración, validación y sistema.

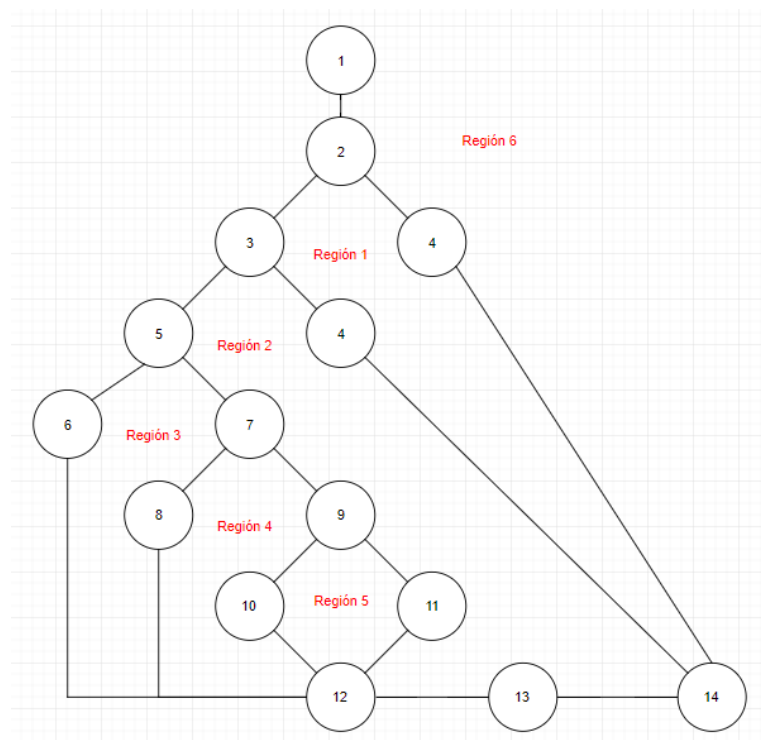
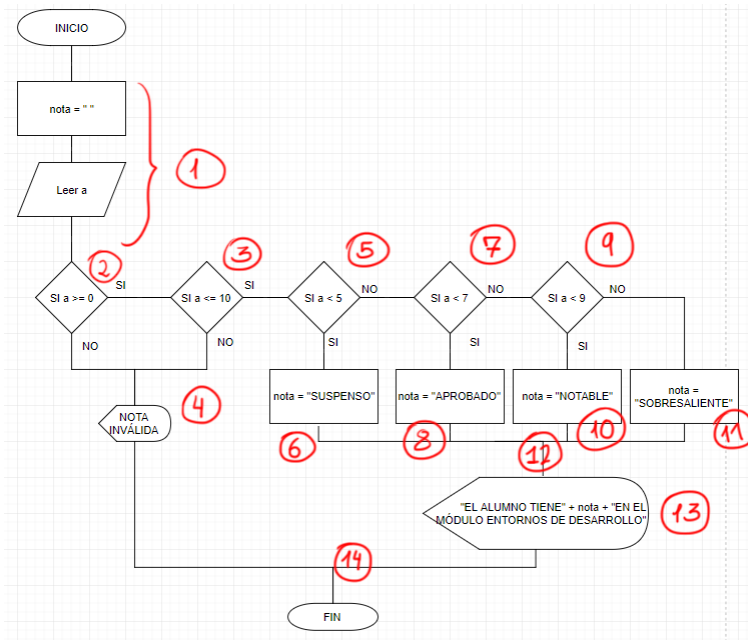
Las pruebas de unidad se centran en comprobar la funcionalidad de cada uno de los módulos.

Las pruebas de integración se realizan cuando se juntan estos módulos, y se comprueba que no haya conflicto entre los módulos y que el diseño es correcto.

Las pruebas de validación se encargan de comprobar que los requisitos del documento ERS coinciden con los desarrollados.

Las pruebas de sistema comprenden diferentes tipos de pruebas, entre las que se encuentran la funcionalidad, el rendimiento, la recuperación y la seguridad entre otras.

3. Realiza el grafo de flujo para el siguiente diagrama de flujo. Indica el número de nodos, aristas, regiones y nodos predicado. Calcula la complejidad ciclomática y muestra los caminos independientes.



Número de nodos $\Rightarrow 15$
Número de aristas $\Rightarrow 19$
Número de regiones $\Rightarrow 6$
Número de nodos predicados $\Rightarrow 5$ (2,3,5,7,9)

Complejidad ciclomática

$V(G) = \text{Número de regiones del grafo} = 6$

$V(G) = \text{Número de aristas} - \text{Número de nodos} + 2 = 19 - 15 + 2 = 6$

$V(G) = \text{Número de nodos predicados} + 1 = 5 + 1 = 6$

Como la complejidad ciclomática está entre 1 y 10, no existe mucho riesgo, el programa es sencillo.

Caminos:

1. 1-2-4-14
2. 1-2-3-4-14
3. 1-2-3-5-6-12-13-14
4. 1-2-3-5-7-8-12-13-14
5. 1-2-3-5-7-9-10-12-13-14
6. 1-2-3-5-7-9-11-12-13-14

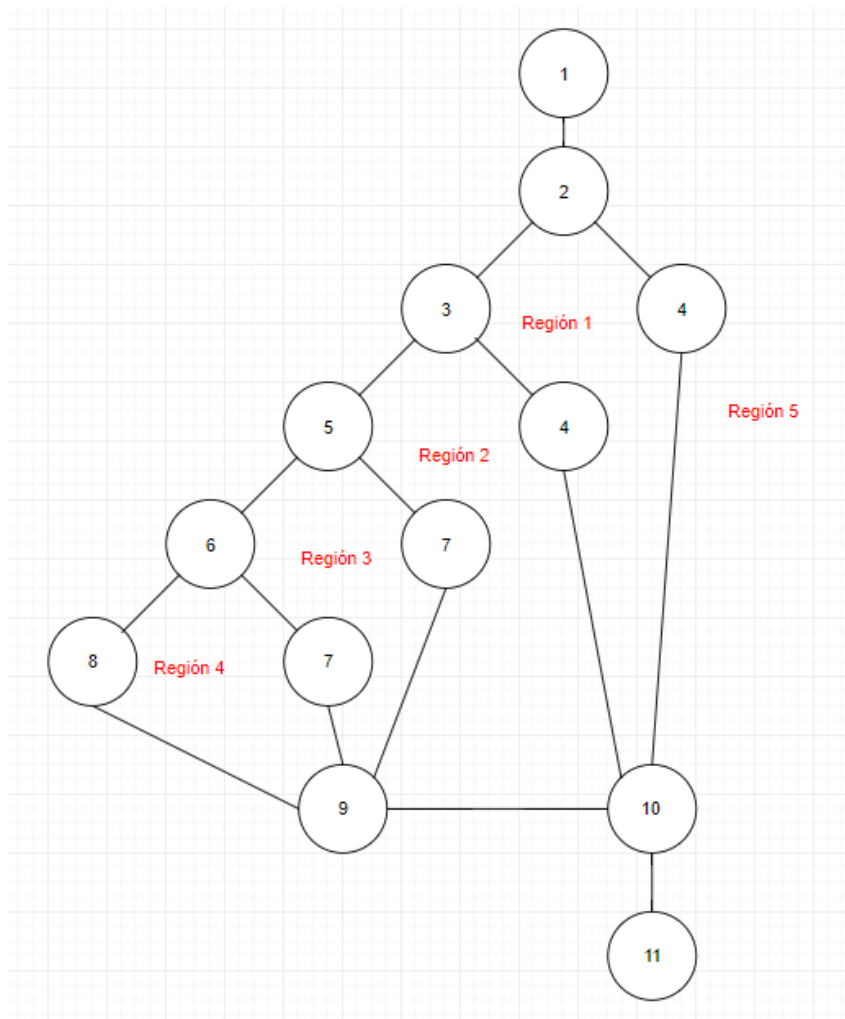
4. Realiza el grafo de flujo para el siguiente código. Indica el número de nodos, aristas, regiones y nodos predicado. Calcula la complejidad ciclomática y muestra los caminos independientes.

```

INICIO
Escribe importe de compra
Leer importe
Escribe mes
Leer mes
SI ((mes = "ENERO") OR (mes = "JULIO")) ENTONCES
    total <- precio * 0'70
SINO
    SI ((mes = "FEBRERO") OR (mes = "AGOSTO")) ENTONCES
        total <- precio * 0'50
    SINO
        total <- precio
    FIN SI
FIN SI
Mostrar total
FIN
  
```

Diagrama de flujo con nodos numerados (1-11) en círculos rojos:

- Nodo 1: Inicio
- Nodo 2: Escribe importe de compra
- Nodo 3: Leer importe
- Nodo 4: Escribe mes
- Nodo 5: Leer mes
- Nodo 6: SI ((mes = "ENERO") OR (mes = "JULIO")) ENTONCES
- Nodo 7: total <- precio * 0'70
- Nodo 8: SINO
- Nodo 9: SI ((mes = "FEBRERO") OR (mes = "AGOSTO")) ENTONCES
- Nodo 10: total <- precio * 0'50
- Nodo 11: SINO
- Nodo 12: total <- precio
- Nodo 13: FIN SI
- Nodo 14: FIN SI
- Nodo 15: Mostrar total
- Nodo 16: FIN



Número de nodos $\Rightarrow 13$
 Número de aristas $\Rightarrow 16$
 Número de regiones $\Rightarrow 5$
 Número de nodos predicados $\Rightarrow 4$ (2,3,5,6)

Complejidad ciclomática

$V(G) = \text{Número de regiones del grafo} = 5$

$V(G) = \text{Número de aristas} - \text{Número de nodos} + 2 = 16 - 13 + 2 = 5$

$V(G) = \text{Número de nodos predicados} + 1 = 4 + 1 = 5$

Como la complejidad ciclomática está entre 1 y 10, no existe mucho riesgo, el programa es sencillo.

Caminos:

1. 1-2-4-10-11

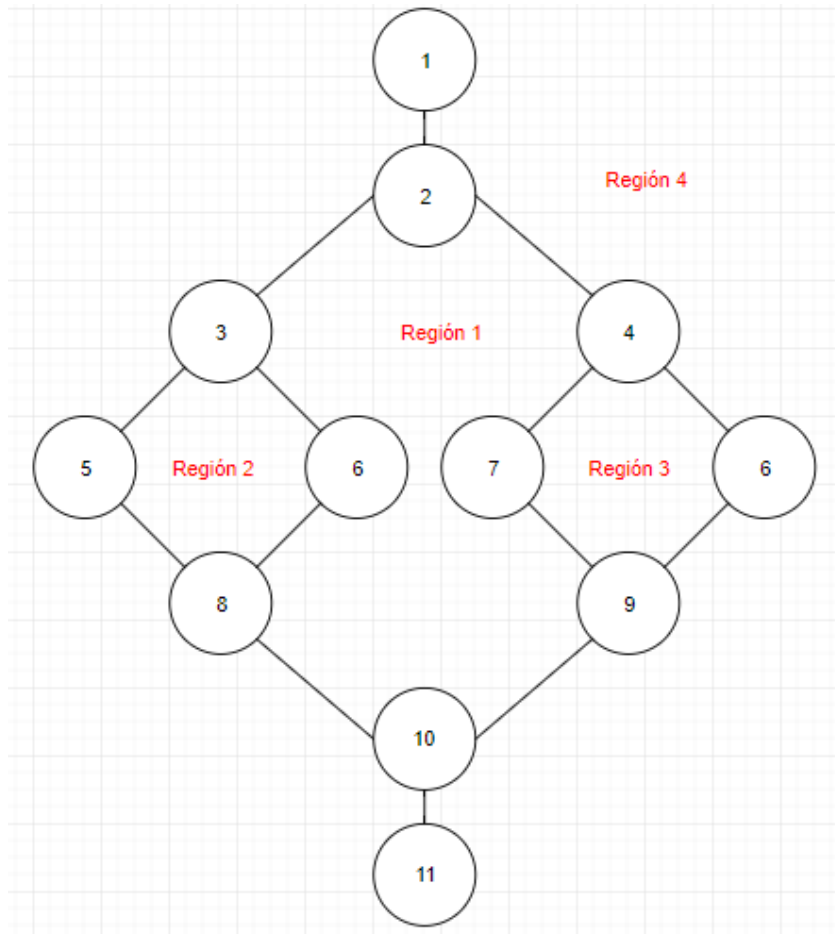
2. 1-2-3-4-10-11

3. 1-2-3-5-7-9-10-11

4. 1-2-3-5-6-7-9-10-11

5. 1-2-3-5-6-8-9-10-11

5. Realiza el grafo de flujo de un programa que recibe tres números por teclado y muestra el menor de los números. Indica el número de nodos, aristas, regiones y nodos predicado. Calcula la complejidad ciclomática y muestra los caminos independientes.



Número de nodos $\Rightarrow 12$
 Número de aristas $\Rightarrow 14$
 Número de regiones $\Rightarrow 4$
 Número de nodos predicados $\Rightarrow 3$ (2,3,4)

Complejidad ciclomática

$V(G) = \text{Número de regiones del grafo} = 4$

$V(G) = \text{Número de aristas} - \text{Número de nodos} + 2 = 14 - 12 + 2 = 4$

$V(G) = \text{Número de nodos predicados} + 1 = 3 + 1 = 4$

Como la complejidad ciclomática está entre 1 y 10, no existe mucho riesgo, el programa es sencillo.

Caminos:

1. 1-2-3-5-8-10-11
2. 1-2-3-6-8-10-11
3. 1-2-4-7-9-10-11
4. 1-2-4-6-9-10-11

¡Buen trabajo!

