

**Statement coverage:** Numero de statements ejecutados (líneas no if ni while)

**Branch coverage:** Condicionales e incondicionales que se cumplen (cumplir las dos opciones de cualquier condicional)

$$\text{Branch Coverage} = \frac{\text{Number of Executed Branches}}{\text{Total Number of Branches}}$$

## ¿Cuántos test debo tener mínimamente para cubrir todos los branches?

$$M = E - N + 2 * P$$

McCabe's Complexity

- M = Complejidad ciclomática.
- E = Número de aristas del grafo. Una arista conecta dos vértices si una sentencia puede ser ejecutada inmediatamente después de la primera.
- N = Número de nodos del grafo correspondientes a sentencias del programa.
- P = Número de componentes conexos, nodos de salida.  
(normalmente 1)

Branch es más confiable para encontrar áreas no testeadas ya que no depende de la cantidad de statements en un camino

**Decision coverage:** condicionales que se cumplen

**Dummy** - Los dummy objects son pasados como argumentos a lo largo del programa pero nunca son usados al final. Usualmente usados para llenar la lista de parámetros.

**Fake** - Los fake objects en realidad son implementaciones funcionales, pero usualmente sus operaciones (métodos) toman atajos (para facilitar el trabajo), pero no pueden ser usados en producción.

**Stubs** - Sus métodos proporcionan respuestas limitadas y acotadas a los test que se realizarán, normalmente no responden nada adicional. Solo a los test para los cuales fueron pensados.

**Mocks** - objetos pre-programados con expectativas y especificaciones a cumplir los test.

**Stub:** Para hacer la prueba más ligera.

**Mock:** Para asegurarse de que los valores sean los esperados.

**Performance testing:** Evalúa la velocidad, uso de recursos, etc.

**Load testing:** Detecta cuellos de botella

**Stress testing:** Puntos de quiebre

**Test Scenario:** Cualquier funcionalidad que se puede probar

**Test-Case:** Conjunto de acciones para verificar una funcionalidad (pasos, datos, etc)

**White-box test:** (Se puede ver dentro del test)

- Statement testing
- Decision testing
- Test Doubles
- Mutation testing
- Search based software

**Black-box test:** (No se puede ver dentro del test)

- Equivalence partitioning
- Boundary value analysis
- State transition testing
- Decision table testing
- Use Case testing

**Equivalence partitioning:** Los datos de entrada se dividen en particiones equivalentes. Hay clases de equivalencia. (ej: edad de 1 a 17 y de 18 a 60).

**Boundary Analysis:** Probar entre valores límites o entre límites de particiones.

**State Transition testing:** Sistema tiene un numero finito de estados y transiciones de uno a otro