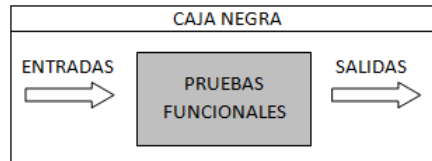


3.C. Caja negra

1. Intruducción

El propósito de las **pruebas de caja negra o funcionales** es comprobar si las salidas que devuelve la aplicación son las esperadas en función de los parámetros de entrada.



Este tipo de prueba, no consideraría, en ningún caso, el código desarrollado, ni el algoritmo, ni la eficiencia, ni si hay partes del código innecesarias, etc. Estos aspectos son comprobados en las pruebas de caja blanca.

Dentro de las pruebas funcionales, podemos indicar los siguientes **tipos**:

- **Particiones equivalentes**: la idea de este tipo de pruebas funcionales, es considerar el menor número posible de casos de pruebas, para ello, cada caso de prueba tiene que abarcar el mayor número posible de entradas diferentes. Lo que se pretende, es crear un conjunto de clases de equivalencia, donde la prueba de un valor representativo de la misma, en cuanto a la verificación de errores, sería extrapolable al que se conseguiría probando cualquier valor de la clase.
- **Análisis de valores límite**: en este caso, a la hora de implementar un caso de prueba, se van a elegir como valores de entrada aquellos que se encuentra en el límite de las clases de equivalencia.
- **Pruebas aleatorias**: consiste en generar entradas aleatorias para la aplicación que hay que probar. Se suelen utilizar generadores de prueba, que son capaces de crear un volumen de casos de prueba al azar, con los que será alimentada la aplicación.
- **Conjetura de errores**: trata de generar casos de prueba que la experiencia ha demostrado generan típicamente errores. En valores numéricos, un buen ejemplo es comprobar si funciona correctamente con el valor 0, ya que si es utilizado como denominador en alguna división podría generar un error en nuestro programa.