

PRUEBA DE ESCRITORIO



Una **PRUEBA DE ESCRITORIO** es la comprobación lógica, de un algoritmo de resolución.

- La **prueba de escritorio** es la herramienta útil para entender que hace un determinado algoritmo, o para verificar que un algoritmo cumple con la especificación sin necesidad de ejecutarlo.
- También sirve para optimizar un algoritmo viendo cómo podemos llegar a un determinado resultado con menos pasos.
- Es una prueba que se realiza de tipo manual y se centra en analizar y evaluar el comportamiento de un programa sin ejecutarlo.

La PRUEBA DE ESCRITORIO permite:

- Detección temprana de errores.
- Ahorro de tiempo y de recursos: Al descubrir y solucionar errores en etapas tempranas del proceso, se evita el desperdicio de tiempo y recursos en la depuración y corrección de problemas más complejos.
- Mejora la calidad del programa.

- Si el programa hace lo que debería hacer
- Si no hace lo que debería hacer, nos permitirá detectar errores como ser:
 - . Si algún paso o instrucción no esta en el orden correcto
 - . Si falta algo
 - . Si algo esta demás
 - . Si los pasos o instrucciones que se repiten lo hacen más o menos veces de lo debido
 - . Si las instrucciones están en un orden apropiado
 - . Otros errores que pueden presentarse

Una **PRUEBA DE ESCRITORIO** es un tipo de prueba algorítmica que consiste en la validación y verificación del algoritmo a través de la ejecución de las sentencias que lo componen (proceso) para determinar sus resultados (salida) a partir de un conjunto inicial determinado de elementos (entrada).

VALIDACIÓN:

¿estamos construyendo el producto correcto?



Consiste en corroborar que el programa satisface las expectativas del usuario

VERIFICACIÓN:

¿estamos construyendo el producto correctamente?



Consiste en corroborar que el programa respeta su especificación

Identificar la porción de código a probar



Analizar la lógica y los requisitos del programa



Seleccionar valores de entrada



Realizar cálculos manualmente



Comparar los resultados esperados con los reales



Corregir discrepancias



Repetir los pasos anteriores con diferentes valores de entrada

Acción Suma es

Ambiente

a,b,s:entero;

Algoritmo

Leer (a);

Leer (b);

S:=a + b

Escribir (s);

Fin Acción

Caso de prueba	Valores de entrada	Resultado esperado	Resultado real	Verificación
1	a = 5, b = 7	12	suma(5, 7) = 12	Correcto
2	a = -3, b = 10	7	suma(-3, 10) = 7	Correcto
3	a = 0, b = 0	0	suma(0, 0) = 0	Correcto

