

Pruebas con Python

Ideas clave

El testing de software o software QA es una disciplina en la ingeniería de software que permite tener procesos de ejecución de un programa o aplicación y una metodología de trabajo con el objetivo de localizar errores de software.

El proceso de testing supone:

- Ahorro de tiempo y esfuerzo.
- Verificar que los errores pasados no se reproduzcan.
- Verificar que el producto sea funcional para el usuario.
- Verificar que no haya ninguna funcionalidad rota en el 'build'.

Existen diferentes tipos de pruebas dependiendo de qué aspecto en concreto se quiera poner a prueba en nuestro programa:

- Prueba unitaria
- pruebas de integración
- pruebas de un extremo a otro (E2E)
- pruebas de aceptación
- pruebas de caja blanca
- pruebas de caja negra
- pruebas de caja gris
- pruebas manuales
- pruebas estáticas
- pruebas dinámicas
- pruebas de la interfaz de usuario
- pruebas de humo
- pruebas de regresión
- pruebas de carga
- pruebas de inserción

Las pruebas unitarias corren pruebas sobre partes del código de forma independiente.

Las pruebas de integración se hacen necesarias cuando queremos probar de una manera sistémica un proyecto como un todo.

Python incluye forma nativa librerías para testing, otras librerías de terceros son:

- Robot
- PyTest
- Prueba de unidad
- DocTest
- Nariz2
- Testify
- Unittest

Con unittest vamos a hacer un proceso de automatización de la verificación de nuestro código.

En unittest no sólo tenemos que comprobar que la función nos devuelva los resultados correctos, también tenemos que asegurarnos de que se disparan las excepciones correctamente según los casos. También podemos comprobar si el tipo de datos que está recibiendo la función mediante parámetro es correcto o no.

Los marcos de prueba como PyTest pueden funcionar directamente con afirmaciones para formar UnitTests en pleno funcionamiento.