# Tipos De Testing

* **Caja Negra**: No está basada en el conocimiento del código o diseño interno, determina la funcionalidad del sistema.
* **Caja blanca**: está basada en la lógica interna de la aplicación y el código. Hace una cobertura de declaraciones del código, ramas, caminos y condiciones.
* **Unitaria**: es la escala más pequeña de la prueba, está basada en la funcionalidad de los módulos del programa, como funciones, procedimientos, módulos de clase, etc.
* **De integración**: se basa en las pruebas de conexiones y comunicaciones entre diferentes módulos. Es esencial en sistemas de cliente\_servidor o red.
* **Funcional**: La caja negra hace la prueba funcional de los requerimientos de la aplicaciones y generalmente se realiza por el programador, en cambio, la prueba funcional es realizada por los testers.
* **De Sistema**: es una prueba de caja negra incluyendo todos los componentes del sistema desde el hardware a la documentación.
* **De aceptación**: es la prueba final basada en las especificaciones del usuario o basada en el uso del programa por el usuario final luego de un periodo de tiempo.
* **De carga**: Está basada en las aplicaciones bajo cargas pesadas, generalmente usadas en sitios web y servidores con gran cantidad de datos donde se determina en cuales puntos existen degradaciones del sistema.
* **De Performance**: Es una de las pruebas finales y sirve para definir los requerimientos y la calidad del software, en base a las pruebas de carga y estrés. Incluye entrevistas con el usuario y programador.
* **De estrés**: Es una prueba de carga y rendimiento o respuesta(performance) basada en la funcionalidad del sistema bajo cargas pesadas, un gran número de repeticiones, manejo de grades datos y demasiadas preguntas a bases de datos grandes.
* **De instalación**: Determina la eficiencia de los procesos que instalan y desinstalan las aplicaciones del programa.
* **De Recuperación**: Es la prueba que evalúa que tan bien se recupera el sistema luego de bloqueos, fallas del hardware u otros problemas catastróficos.
* **De seguridad**: evalúa que tan bien el sistema se protege contra accesos, internos o externos, no autorizados, esta prueba requiere sofisticadas técnicas y herramientas.
* **De Compatibilidad**: Evalúa el desempeño del software en diferentes hardware, sistemas operativos, redes, etc.
* **De Exploración**: es una prueba informal del software que no está basada en ningún plan o caja de prueba y a menudos los testers aprenden del programa al explorar todas las aplicaciones posibles.
* **Alfa(Alpha):** es la prueba cuando la aplicación está cerca de la entrega al usuario. Se hacen pequeños cambios generalmente en el diseño de interfaces. Esta prueba es hecha por usuarios.
* **Beta**: Es la búsqueda de defectos en el programa completo. Generalmente es hecha por usuarios.
* **De Mutación**: esta prueba está basada en la introducción deliberada de diferentes códigos externos al programa (defectos) para reexaminar si estos defectos pueden ser detectados. Requiere gran disponibilidad de recursos de computación.
* **Conjetura de errores**: Se enumera una lista de posibles equivocaciones típicas que pueden cometer los desarrolladores y de situaciones propensas a ciertos errores, Es recomendable imaginar que el programador pudiera haber interpretado algo mal en la especificación. También interesa imaginar lo que el usuario puede introducir como entrada a un programa. El valor cero es una situación propensa a error tanto en la salida como en la entrada.